Alma Mater Studiorum · University of Bologna

School of Science
Department of Physics and Astronomy
Master Degree in Physics

Modeling and data analysis of biochemical oscillators using Chemical Master Equation and AI: applications to the NF-kB activity in patient derived xenografts

Supervisor:
Prof. Enrico Giampieri

Co-supervisor:
Prof. Daniel Remondini

Submitted by:
Manuela Carriero

Academic Year 2022/2023

"Behind the great achievements of a Gauss or an Einstein is in all cases a life devoted to contemplation, curiosity, collaboration and, perhaps most of all, hard work. "[1]

Brian D. Burrell, Professor of Neuroscience

[1]GENIUS IN A JAR, Scientific American 2015 Sep; 313(3):82-7.

DEDICATION

Dedicato alla mia famiglia

Abstract

There are two stages from DNA sequence of a gene to protein: transcription, i.e. the process of making a strand of RNA molecule, and translation, that is the process by which a protein is synthesized from the information contained in a molecule of RNA. In the process of transcription, proteins called "transcription factors" play a central role because they bind to a DNA sequence and help the transcription initiation complex. In this work of thesis, we are particularly interested in modeling the nuclear factor-kappa B (NF-kB) activity which is ubiquitous within cells and its dysfunction leads to chronic diseases, cancers, neurodegenerative diseases and much other. However, we do not start immediately modeling its behavior. In this stochastic context, firstly we aim to deepen into algorithms to solve the Chemical Master Equation (CME) giving an our alternative algorithm called "hybrid" because it combines the Gillespie's Stochastic Simulation Algorithm (SSA) with the tauleaping algorithm with the aim to improve the algorithm's speed; secondly we analyse the stochastic simulation results of three basics genetic circuits (the simplest model of gene expression, the autorepressor and the toggle switch); third, we faced the problem of parameters estimation of these simple models using artificial neural networks (ANN); finally, aware of what we have learned after all such steps, we provide a very little NF-kB model using the CME. The relevant results are the following: the hybrid algorithm applied to the first genetic model is faster than the SSA in configurations where the number of molecules produced tends to be high; periodicity arises from what we defined as unpredictable (being stochastic processes), particularly from the autorepressor using wavelet transform; ANN learn to predict parameters given autocorrelations as input, providing also information about the chemical species; finally, our little NF-kB model shows an oscillating behavior as expected by experiments.

Contents

Part I

Introduction into biological stochasticity and physical methods for its investingation

Chapter 1

Introduction

"There's no gene for fate"

from GATTACA film

Kevin Burrage et al. in their important paper *Stochastic simulation in systems biology* [12] highlight how mathematical modelling and computational simulation perform essential roles in biology. Models are useful for many different purposes: perhaps the most important of these are crystallising our assumptions and testing them, guiding experiments and looking at experimentally unreachable scenarios, as well as the great ability to make new predictions.

Ideally, there should be a virtuous cycle between experiment and theory to understand the natural system in which we are interested.

If the model and experiments agree, then parameters for the model can be inferred from the data and the model refined, again leading to further experiments. If they do not agree, this implies that the hypothesis was inappropriate: the model needs revision, and theoretical tests using different models can provide a starting point for further experiments.

As defined by Kevin Burrage et al., model is an abstract representation of the system in which we are interested and this is usually formulated mathematically. This model can, in simple cases, be solved analytically for a particular question, giving us an *exact* analytical solution to the question posed to the model. This means that the full probability distribution for the state of the biological system over time can be calculated explicitly.

In general, however, analytical solutions for all but the most simple models do not exist, and we need to use methods to solve the model numerically. These last methods give rise to numerical solutions which are approximations to the analytical solution.

Numerical solution will try to mimic the behaviour of the real system over time. This is also refered to as a simulation.

An important point to remember when we talk about mathematical models is that, in a sense, "all models are wrong" but this does not mean their conclusions or predictions are false, as the famous quote continues, "but some are useful".

Rather, we should bear in mind that all models are abstractions of reality, simplified versions of the real systems that they represent. A model is a vehicle for gaining understanding, and we would not gain any new understanding from a model that was as complicated as the real system.

Almost all models are phenomenological; that is, they are based on a set of simplified assumptions that we have distilled from the real system using intuition rather than rigorous proof from basic principles. Incorporating only the important ones allows us to crystallise our understanding of the system (and if the model gives wrong results, we know that our assumptions were wrong or incomplete, hence it is in any case a gain in knowledge of our system under study).

Creating phenomenological models is not a trivial task: "sensing which assumptions might be critical and which irrelevant to the question at hand is the art of modeling". The choice of what to include in the model lies with the modeller and his aims.

1.1 Heterogeneity in biology

Stochastic computational methods have become commonplace in science because they are able to appropriately account for one key feature in biology: this is heterogeneity.

There are three different sources of heterogeneity in natural systems: genetic, environmental (i.e. also known as *extrinsic noise*) and stochastics (i.e. also called *intrinsic noise*).

Genetic heterogeneity occurs through the production of single or similar phenotypes through different genetic mechanisms [42]. The general "rule" in biology is that cells and animals with different genes should clearly be different in phenotypes. However, the so called "convergent evolution" represents an exception to this rule where different species evolved similar phenotypic features independently. This phenotypic similarity is thought to usually have a different genetic basis, showing that the relationship between genotype and phenotype is not a simple, linear one [43].

On the other hand, even isogenic (that is, genetically identical) organisms or populations of cells can be very different; the famous cloned animals are excellent illustrations of this phenomenon, known as non-genetic or phenotypic heterogeneity. Daniel Stockholm et al. in their work named *The Origin of Phenotypic Heterogeneity in a Clonal Cell Population In Vitro*[44] show phenotypic switch in initially identical cells. This suggests that heterogeneity is not only generated by genetic variation but it is triggered by two other factors.

The first one is the consequence of *extrinsic factors*: initially identical cells may become different because they encounter different local environments that induce adaptive

responses. Examples of each of these factors could be: if we were interested in animal populations, the weather an animal experiences in a particular year or if we were interested in levels of protein expression of a cell population, we might look at differences in individual cells such as ribosome number and cell cycle stage.

The third source of heterogeneity is instead *intrinsic* to cells and may occur even in homogeneous environments. Indeed identical genotype and environmental exposure are not sufficient to guarantee a unique phenotype. The intrinsic heterogeneity is present in all living (and non-living) systems but often masked by the macroscopic scale at which we observe them. Spudich and Koshland were perhaps the first to observe this in a cell biology context. As explained in their paper *Non-genetic individuality: chance in the single cell* [45], they noticed that individual bacteria from an isogenic population maintained different swimming patterns throughout their entire lives. This was a visible manifestation of the third source of heterogeneity: *chance*, otherwise known as *stochasticity* or *intrinsic heterogeneity* (often interchangeably called *intrinsic noise*).

This arises from random thermal fluctuations at the level of individual molecules. It affects the DNA, RNA, protein and other chemical molecules inside cells in many ways, most notably by ensuring that their reactions occur randomly, as does their movement (via Brownian motion). Very clear examples could be the following: consider a single mother cell dividing into two daughter cells of equal volume. During the division process, all the molecules in the mother cell are in Brownian motion according to the laws of statistical mechanics. The probability that each daughter cell inherits the same number of molecules is infinitesimally small. Even in the event that the two daughter cells receive exactly one copy of a particular transcription factor, each transcription factor will perform a Brownian random walk through its cellular volume before finding its target promoter and activating gene expression. Because Brownian motion is uncorrelated in the two daughter cells, it is statistically impossible for both genes to become activated at the exact same time, further amplifying the phenotypic difference between the two daughter cells. These are just two examples of the many sources of gene expression variability that arise in isogenic cells exposed to the same environment [46].

Intrinsic heterogeneity is inherent to the process of gene expression and cannot be predicted (except in a statistical sense) or fully eliminated. Thus two identical genes in an identical intracellular environment would still be expressed differently, even in the absence of the previous two sources of heterogeneity.

Hence, the reason of terminologies is clear: we talk about extrinsic heterogeneity because it arises from other, outside, sources and affects all genes inside a cell equally, whereas the random fluctuations are actually inherent to the expression of a single gene.

It is important to point out that these three sources of heterogeneity are not independent because they are all interconnected in their effects on cells and populations. For instance, cellular decision involves both environmental and intrinsic heterogeneity, the former in affecting which stable states are possible and the latter in switching between them.

Another example is that of genetic mutations, where intrinsic noise causes the mutations that then contribute to genetic heterogeneity and so causing genetic mutations that are essential for creating the heritable variation, thus allowing evolution to occur.

From this, a very interesting example of the interplay of all three types of heterogeneity is evolution, one of the most fundamental processes in nature: evolution acts on phenotypes, which have a genetic basis but are also affected by both extrinsic and intrinsic noise. Studying how evolution shapes noise is a key challenge and this is highlighted in the work *Adaptive noise* by M. Viney et al. [47]. If our DNA is much different from the first humans appeared on Earth (and so also our phenotype), is due to all this complex interplay among sourcies of heterogeneities.

Finally focusing on intrinsic heterogeneity, we have to point out how this can be a double-edged sword: intrinsic noise has both positive and negative influence that have to be accounted for cells and organisms. One positive effect is already mentioned: it allows evolution to occur. One other related example is that some biological systems have evolved to make use of it: for instance, in the case of persister-type bacteria, which form a subset of some bacterial populations and can withstand antibiotic treatments even though they do not have genetic mutations for resistance. This is an example of cellular decision making, the ability of cells to randomly transition between different stable, heritable states.

Some of the most well-known examples of cellular decision making are bacteria, and yeast, which are easier to investigate experimentally than large multicellular organisms. Yet, this phenomenon is common to cells at all levels of life and is argued to be one of the key processes in cellular development. The main difference is that in unicellular organisms, cellular decision making is useful for cheap, non-genetic adaptation in the face of fluctuating environments, whereas in multicellular organisms it is used to produce distinct cell types and functions within a constant environment.

Intrinsic noise is also an interesting object of study since it represents a "fingerprint" with which to explore fundamental questions on mechanisms and dynamics of gene regulation by means of measurements of cell-to-cell variability in gene expression [46].

However, as said before, intrinsic noise causes mutations which can also have negative effects. Deleterious mutations that are phenotypically expressed frequently result in loss of important functions or even organismic death, and it has been shown that a substantial proportion of mutations are deleterious. Mutations are also thought to cause cancer, which has become a serious disease in modern times.

1.2 Motivation of the work and the consequent workflow

Following the line of reasoning of the previous section, cancers like colonrectal cancers (CRC) are mainly caused by mutations that target oncogenes leading to novel or increased functions, or alterations that drive to loss of functions of tumor-suppressor genes. This provides neoplastic cells with a survival advantage over the surrounding normal intestinal epithelium.

Approximately 70% of CRC cases are sporadic and, generally, develop from a point mutation that occurs spontaneously during lifetime.

It is very likely that these mutations lead to a malfunction of NF-κB transcription factor activity, whose behavior is modulated by its regulators, of the proteins family IκB, but that in CRC it is constitutively expressed.

The biology of NF-κB will be explained in Chapter 8 and then a simple model of the feedback loops which involve him will be furnished. However before arriving to this point, we will examine simple genetic models that will serve as a "training" before diving into more complex stochastic models.

This will let us to explore *methods* firstly for stochastic processes simulations, namely we will review them in Chapter 3 and provide an our alternative algorithm called "hybrid" in section 3.4.3 in order to numerically solve the Chemical Master Equation with the aim to provide an algorithm that is faster than the SSA (described in section 3.4.1) but that preserves the accuracy in results; secondly, methods for analyzing data coming from these stochastic simulations in Chapter 5, in particular deeping into the use of Continous Wavelet Transform that they turn out to be an interesting "instrument" for the analysis of stochastic processes signals (someone calls them also as "a mathematical microscope for scanning signals"[30]); and last, but not least, a method to extrapolate model paramaters that in this thesis is the Artificial Neural Networks (their basics principals are explained in Chapter 6 and their application in Chapter 7). About this last point, we thought to give the autocorrelation values as input because they are dependent on model parameters, hence we train ANNs to recognize the model parameters (that are in this case the constant rates of reactions, which are explained in Chapter 2). Nowadays neural networks are used in a lot of fields and so their use in "absolute terms" is not a great novelty. The innovation lies in the context in which they are used because typically the problem of parameters estimation is faced using standard methods such as Maximum Likelihood Estimation (MLE) that have several disadvantages. Hence, we tried an alternative method.

Finally a little NF-κB model will be provided with the use of Gillespie's Stochastic Simulation Algorithm to simulate its activity (Chapters 9 and 10).

1.3 Deterministic versus stochastic models

Before the discovery of intrinsic noise, mathematical and computational methods that have traditionally been used were typically *systems of differential equations* which are both *continous* and *deterministic*, i.e. their state variables are real numbers representing the concentrations of molecules and they do not include noise.

Such models can be considered as accurate only when we are interested in the mean dynamics of a large number of molecules, large enough that we need not worry about individual molecules but can approximate them as concentrations [12]. Above a molecular population size of the order of Avogadro's number, the fluctuations from intrinsic noise are averaged out and the deterministic approximation becomes increasingly valid.

This is because intrinsic noise, as a rule of thumb, behaves as $\frac{1}{\sqrt{X}}$, where X is the number of molecules in the system. This is confirmed by experimental studies that find that total noise does scale roughly as the inverse square of abundance until high abundances. At this point, extrinsic noise is thought to take over as the dominant source of noise [48].

Therefore, it often becomes necessary to include the effects of stochasticity in biological models, especially for small systems with low populations of some molecular species, such as *gene expression networks*. Here, *discrete stochastic models* must be used, whose variables represent actual molecular numbers.

In the cellular environment, the reactant numbers tend to be of the order 10-1000, hence a reaction altering the population by one or two generates a large relative change, and the molecule numbers no longer evolve differentiably. Furthermore, reactions no longer occur "continuously" over an infinitely small time interval, but rather progress in a series of steps of finite time width.

A very clear example may be to imagine the national birth rate as compared to the chances my next-door neighbor will have a baby. One often hears statements such as: "Every X minutes, a baby is born in the US." That clearly cannot be true of my next-door neighbor. Evolution of the population of an entire country can be well-described using differential equations, but individual courtship is an essentially probabilistic affair, requiring a more sophisticated formulation [49].

Thus, for many reactant molecules, the species concentrations evolve both continuously and differentiably. When small numbers of reactants are involved, due to the probabilistic nature of individual reaction events and the finite change in molecule numbers incurred, the concentration evolves step-wise.

And so, while *the law of mass balance* is used to construct deterministic chemical rate equations which take the form of a system of coupled nonlinear differential equations with the assumption that the concentration of the reactants varies both continuously and differentiably (i.e. in case of molecule numbers of the order 10^{23} where a change of one or two molecules in a population of 10^{23} is, for all intents and purposes, infinitesimal), *the probability balance* is used as conservation law of this microscopic description. For

such purpose we will consider the *Chemical Master Equation* which is used to describe the distribution of a population of stochastic systems across states.

1.4 The importance of random numbers generation

Before starting, let us briefly make clear the importance of random numbers generations in this context. Random number let us simulate what we have called "biological stochasticity". We will see very well this when talking about stochastic simulations in biological systems in Chapter 3. But we will exploit them also for the needing of a particular distribution of parameters for the same model (that is made in Chapter 6). Thus, for simulations, it is a particular important key.

Chapter 2

Modeling kinetics reactions in deterministic framework

In this chapter, we describe three simple genetic circuits that are common in nature: the simplest process of gene expression and protein synthesis, the autorepressor case and the toggle-switch model. Hence we need to describe different chemical reactions.

Generally, we can have a chemical reaction like this:

$$\alpha \mathrm{A} + \beta \mathrm{B} \ldots \rightleftharpoons \sigma \mathrm{S} + \mathrm{rT} \tag{2.1}$$

where A and B are the reagents and S and T are the products of the reaction. This reaction is reversible and, as such, according to the *Law of Mass Action*, we can define a *forward reaction rate* and a *backward reaction rate*. The reaction rates are related to a constant, for instance the *forward reaction rate* is related to k_+, and to the concentration of reagents:

$$rate = k_+ A^\alpha B^\beta \tag{2.2}$$

where α and β are called the stechiometric coefficients. Analogously, we can write an equation for the backward reaction rate.

However, the specific forms of these equations rates depend on the order of reaction. In the listed genetic circuits that we will describe, we will consider only first order reaction rates.

When we have a chemical reaction, we are interested in describing it by using differential equations, hence in a deterministic framework.

Thus we will turn these biological processes into a deterministic mathematical description, but then the molecular noise in these systems will be taken into account and we will study them by means of stochastic methods.

2.1 Constitutive gene expression

In the simplest possible model of *constitutive* gene expression, a RNA molecule is produced at a constant rate k_1 and destroyed in a first-order reaction with rate constant k_2. Fig. 2.1 shows a sketch of this simple biological process.

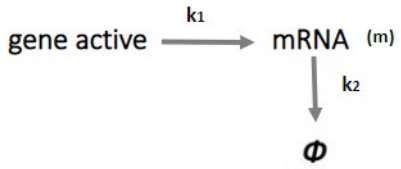

Figure 2.1: Schematic of a constitutive gene expression model with transcription rate k_1 and mRNA degradation rate constant k_2.

As already anticipated in the introductory part, if the total number of a particular transcript m is large, the kinetics is conveniently represented as a deterministic differential equation for species concentrations. This plays the role of the mass-balance conservation law, i.e. change = flux in - flux out [46].

$$\frac{dm}{dt} = k_1 - k_2 m \tag{2.3}$$

This approximation breaks down in cells when the copy numbers of transcripts are small with the need of a probabilistic reformulation in which the conservation law is described by the Chemical Master Equation explained in the next Chapter 3.

In the constitutive expression model, transcript births and deaths occur as uncorrelated events, such that in any short time interval, dt, the probability of one transcript production is $k_1 dt$, and the probability of one transcript degradation is $k_2 m dt$ [46].

The scheme in Fig. 2.1 shows only the transcription process in case gene is always active.

We can look at this model also considering protein production as shown in Fig. 2.2.

Similarly to Eq. 2.3, we can write the rate of change of protein concentration p as:

$$\frac{dp}{dt} = k_3 m - k_4 p \tag{2.4}$$

2.2 Regulated gene expression

Many genes are constitutively expressed at intermediate levels, which represents a permanent cost but provides an immediate benefit when the protein is needed. On the other hand, *regulated* genes are only expressed under certain necessary conditions in order to

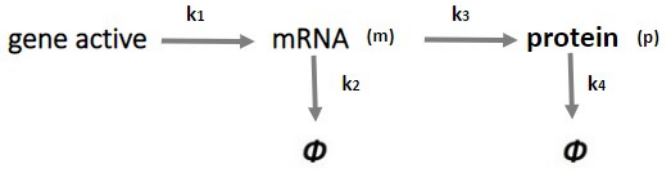

Figure 2.2: Schematic of a constitutive gene expression model with transcription rate k_1 and mRNA degradation rate constant k_2, translation rate k_3 and protein degradation rate constant k_4.

save cellular energy. In other words, they are switched on only when needed and they can be termed also as "inducible". Gene regulation is the key ability of an organism to respond to environmental changes [52]. Fig. 2.3 shows a sketch of gene regulated protein synthesis model.

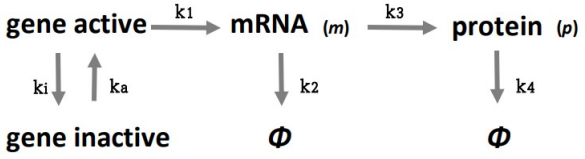

Figure 2.3: Schematic of a regulated gene expression model with gene activation rate k_a and gene inactivation rate k_i, transcription rate k_1 and mRNA degradation rate constant k_2, translation rate k_3 and protein degradation rate constant k_4.

In particular, we can consider a two-state model of gene expression.

This model considers two promoter states: an OFF state, in which no transcription occurs and an ON state, which has transcription rate k_1.

The constants k_a and k_i define the transition rates between the two states, and k_2 is a first-order rate constant for transcript degradation.

The OFF state is usually associated with a closed chromatin state in which the binding sites for transcription factors are inaccessible, whereas the ON state is associated with the open active chromatin state [46].

And so, a simple way to include transcription factor control in the kinetic equations above is to modify k_1 using the promoter activity function $g(\cdot)$. Hence, the deterministic equation Eq.2.3 written for RNA in case of unregulated gene becomes:

$$\frac{dm}{dt} = k_1 \cdot g(\cdot) - k_2 m \tag{2.5}$$

while the Eq.2.4 for proteins keeps unchanged.

$g(\cdot)$ can have different behaviors: in the case shown in Fig. 2.3, it makes the gene alternating between active and inactive state.

2.3 The Autorepressor model

As shown schematically in fig. 2.4, in this case gene product represses its own transcription.

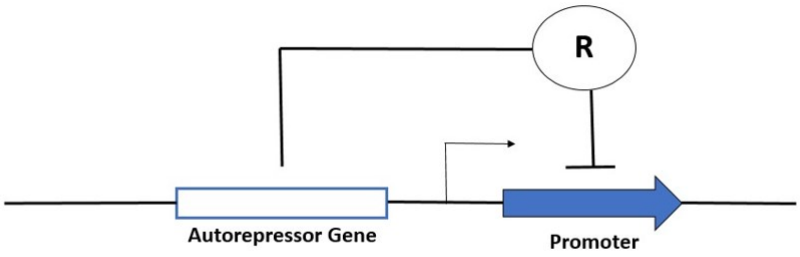

Figure 2.4: A schematic of an auto-repressing circuit where a blunt line indicates a repressing action while a connector ending with an arrow indicates an activating action.

Proteins produced by the gene decreases the rate of activation of the gene. Eq.2.5 and Eq.2.4 written for the first simple protein synthesis model are then modified in the following way:

$$\frac{dm}{dt} = k_1 \cdot g_R(r) - \beta_m m \tag{2.6}$$

as regards the rate of change of RNA molecules. Analogously, for proteins we can write the same equation as Eq. 2.4 written for constitutive expression and for the regulated gene case but, for the sake of clarity, we denote p as r in order to highlight that protein product is behaving as autorepressor. Hence:

$$\frac{dr}{dt} = k_3 m - k_4 r \tag{2.7}$$

2.4 The Toggle-Switch model

The genetic toggle-switch is a system of two mutually repressing genes. A simplified sketch of this model is shown in Fig.2.5.

The repression of gene 2 is mediated by an inducer (related to gene 1) that regulates gene expression. Similarly this happens for gene 1 repression.

We would expect the system to exhibit *two* mutually exclusive behaviors: either r_1 is high, keeping expression of r_2 low, or conversely, r_2 is high, keeping expression of r_1 low [49].

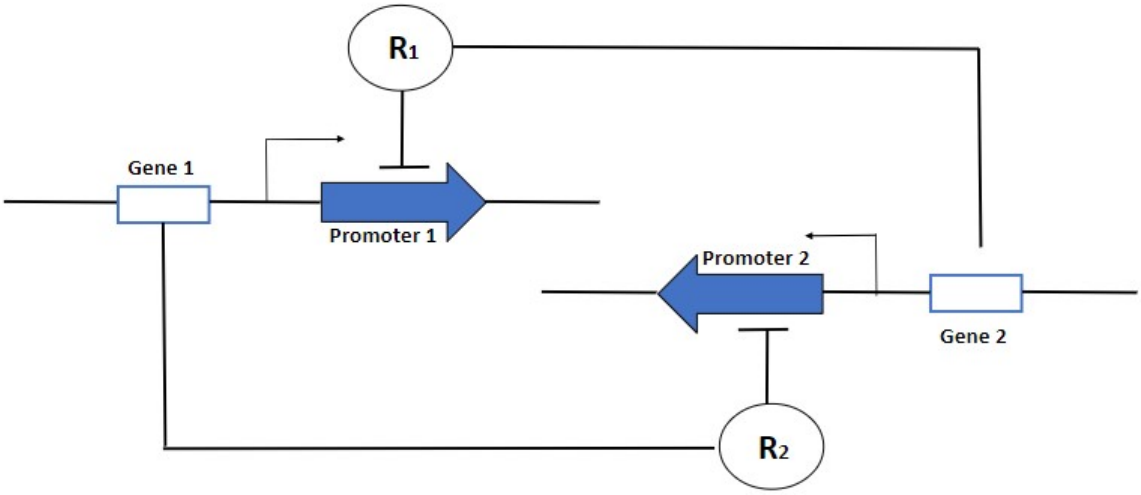

Figure 2.5: A schematic of the toggle switch where a blunt line indicates a repressing action while a connector ending with an arrow indicates an activating action.

For simplicity, we assume that the switch is composed of symmetric elements so that the rate constants are identical for each half of the network. Then, as seen in the other cases, the mathematical model takes the form:

$$\frac{dm_1}{dt} = k_1 \cdot g_R(r_2) - k_2 m \qquad \frac{dm_2}{dt} = k_1 \cdot g_R(r_1) - k_2 m \tag{2.8}$$

$$\frac{dr_1}{dt} = k_3 m_1 - k_4 r_1 \qquad \frac{dr_2}{dt} = k_3 m_2 - k_4 r_2 \tag{2.9}$$

Chapter 3

Stochastic simulation in systems biology

3.1 Stochastic Process

In a "rough" sense, a random process is a phenomenon that varies to some degree unpredictably as time goes on. If we observed an entire time-sequence of the process on several different occasions, under presumably "indentical" conditions, the resulting observation sequences in general, would be different [57].

This "rough" definition remembers the definition of stochasticity given for isogenic cells in the same environment.

In the following sections, we will define more rigously what a stochastic process is by using the Van Kampen notations written in his *Stochastic Processes in Physics and Chemistry* book [7] and some possible tools to study random processes.

3.1.1 Definition

A "random number" or "stochastic variable" is an object X defined by: a set of possible values (that we can call also "set of states" or "sample space") and a probability distribution over this set. Once a stochastic variable X has been defined, an infinity of other stochastic variables derives from it, namely all quantities Y that are functions of X by some mapping f. These quantities Y may be any kind of mathematical object, in particular also functions of an additional variable t,

$$Y_X(t) = f(X, t) \tag{3.1}$$

Such a quantity *Y(t)* is called a "random function", or, since in most cases t stands for time, a *stochastic process*. Thus a stochastic process is simply a function of two variables, one of which is the time t, and the other a stochastic variable X. On inserting for X one

of its possible values x, we obtain an ordinary function of t:

$$Y_x(t) = f(x,t) \tag{3.2}$$

called a *sample function* or *realization* of the process. In physical parlance one regards the stochastic process as an "ensemble" of these sample functions.

3.1.2 Markov Processes

There is a subclass of stochastic processes that have the Markov property. Such processes are by far the most important in physics and chemistry. A Markov process is defined as a stochastic process with the property that for any set of n *successive* times (i.e., $t_1 < t_2 < ... < t_n$) one has

$$P(y_n, t_n | y_{n-1}, t_{n-1}, ..., y_1, t_1) = P(y_n, t_n | y_{n-1}, t_{n-1}) \tag{3.3}$$

that is, the conditional probability density at t_n, given the value y_{n-1} at t_{n-1}, is uniquely determined and is not affected by any knowledge of the values at earlier times. The system loses any kind of information of its state before the present value, and so the markovian systems are usually called memory-less.

3.2 The Chemical Master Equation (CME)

As we have understood, stochasticity is very important in our biological context. The master equation approach is used to describe the time evolution of the probability of a stochastic system to be in a specific configuration in the framework of Markov processes. Indeed, the mathematical derivation of the Master equation starts from the equation (3.3), since we can write such relation in a more formal way [5], that gives the probability of a transition of the system of going to the state y_3 from the state y_2, that is at the time $t+\tau$, given the observation of the state y_1 at the time t. Given that T_τ is the *transition probability* that does not depend on the moment in time but only on the elapsed time and calling $P(y_2, t+\tau | y_1, t) = T_\tau(y_2|y_1)$, we obtain the important Chapman-Kolmogoroff equation for the transition propensity:

$$T_{\tau+\tau'}(y_3|y_1) = \int T_{\tau'}(y_3|y_2) T_\tau(y_2|y_1) dy_2 \tag{3.4}$$

The CK equation for T_τ is a functional relation, which is not easy to handle in actual applications. The master equation is a more convenient version of the same equation: it is a differential equation obtained by going to the limit of vanishing time difference τ'. Hence, taking the first order term of the Taylor series of the $T_{\tau'}(y_3|y_2)$ integral for small τ', we can write it as:

$$T_{\tau'}(y_3|y_2) = (1 - a_0\tau')\delta(y_3 - y_2) - \tau' W(y_3|y_2) + O(\tau'^2) \tag{3.5}$$

Here $W(y_2|y_1)$ is the *transition probability per unit time* from y_1 to y_2 and hence $W(y_2|y_1) \geq 0$. The coefficient $1 - a_0\tau'$ in front of the delta function is the probability that no transition takes place during τ', hence $a_0(y_1) = \int W(y_2|y_1)dy_2$.

Now we insert this expression in the CK equation for $T_{\tau'}$

$$T_{\tau+\tau'}(y_3|y_1) = [1 - a_0\tau']T_\tau(y_3|y_1) + \tau' \int W(y_3|y_2)T_\tau(y_2|y_1)dy_2 \tag{3.6}$$

Divide by τ', go to the limit $\tau' \to 0$, and use $a_0(y_3) = \int W(y_3|y_2)dy_2$:

$$\frac{\partial}{\partial\tau}T_\tau(y_3|y_1) = \int W(y_3|y_2)T_\tau(y_2|y_1) - W(y_2|y_3)T_\tau(y_3|y_1)dy_2 \tag{3.7}$$

This is the differential form of the Chapman–Kolmogorov equation that is known as the *Master equation*. It can be written in the simplified and more intuitive form

$$\frac{\partial P(y,t)}{\partial t} = \int W(y|y')P(y',t) - W(y'|y)P(y,t)dy' \tag{3.8}$$

This can be recognized as an influx of probability to the state y from all the "surrounding" (in the sense connected) states y' and an efflux from y to every state y' to which it can move to [5]. This thesis is concerned with solving the master equation for chemical systems and if the system state space is discrete, as when we work with a system with a discrete number of individuals or molecules, we can write the probability as $P_n(t)$ to represent the *discreteness* of the state space. In this case the master equation can be called *Chemical Master Equation* (referring to a chemical environment) and it will be written with sums instead of integrals [5]:

$$\frac{dP_n(t)}{dt} = \sum_{n'}[W_{n,n'}P_{n'}(t) - W_{n',n}P_n(t)] \tag{3.9}$$

as well as $W(y'|y)$ represents the transition probability per unit time from y to y' in the continuous equation, $W_{n,n'}$ is the transition probability per unit time from the state n to the state n' in the discrete equation. Hence, also in this form, we can recognize the meaning of Eq 3.9 as we did for Eq 3.8 and state that *the master equation is a gain–loss equation for the probabilities of the separate states n* [21]. The first term is the gain due to transitions from n' to n states, and the second term is the loss due to transitions from n to n' states.

The Eq 3.9 shows a linear relationship between transition probabilities, therefore we can write it as a linear dynamical system gaining a more compact form of the CME:

$$\partial_t\vec{P}(t) = \mathbb{W}\vec{P}(t) \tag{3.10}$$

where $\vec{P}$ is a column vector with components p_n and $\mathbb{W}$ is called the *transition matrix* because it is the matrix of the transition rates. $\mathbb{W}$ is defined as

$$\mathbb{W} = \begin{cases} \mathbb{W}_{n,n'} \text{ for } n \neq n' \\ \mathbb{W}_{n,n} = -\sum_{n \neq n'} W_{n',n} \end{cases} \tag{3.11}$$

In the general case, the matrix $\mathbb{W}_{n,n'}$ should obey the following properties:

$$\mathbb{W}_{n,n'} \geq 0 \text{ for } n \neq n' \tag{3.12}$$

$$\sum_n \mathbb{W}_{n,n'} = 0 \text{ for each } n' \tag{3.13}$$

The Eq 3.13 states that the matrix $\mathbb{W}$ has zero determinant. We can have a confirmation of this property by writing $\mathbb{W}$ for $N = 3$, i.e. the number of possible states are $n \in [0, 3]$, that we can do as an exercise:

$$\mathbb{W} = \begin{pmatrix} -(W_{2,1} + W_{3,1}) & W_{1,2} & W_{1,3} \\ W_{2,1} & -(W_{1,2} + W_{3,2}) & W_{2,3} \\ W_{3,1} & W_{3,2} & -(W_{1,3} + W_{2,3}) \end{pmatrix} \tag{3.14}$$

A zero determinant means that there is a zero eigenvalue and the eigenvector corresponding to the zero eigenvalue is the *stationary distribution* $P(n)$, the distribution to which the stochastic process always converges, i.e. $\vec{P}(t) = 0$, as long as the transition propensities $W_{n,n'}$ are not a function of time.

On the other hand, one can solve the master equation to obtain n solutions $P_n(t)$ that depend on time since, in general, the master equation represents a set of n deterministic equations, with n that is the number of states and so, in our context, the number of molecules. In this case, bear in mind a fundamental property of the master equation: as $t \to \infty$ all solutions tend to the stationary solution.

If the system is fully connected (cannot be broken into two non communicating pieces) the stationary distribution is guaranteed to be unique [5]. The stationary distribution will be obviously positive, i.e. all its terms are with positive sign and the sum of all its components is 1 (being a probability distribution). All the other eigenvalues will be with negative module, and the corresponding eigenvectors will have total sum of the components equal to zero, as they can be interpreted as the difference between the present distribution and the stationary one, both having total sums of the components equal to 1. A special role is played by the eigenvalue with the smallest absolute value, which it means that its eigenvector is the longest-standing one. This eigenvector is referred as the *metastable* state and its eigenvalue gives a time-scale of the time of convergence to the stationary distribution.

We end up this section contemplating the impressive similarity between the Chemical Master Equation and the Shrödinger's equation: just as the solution of the Shrödinger's

equation is the probability distribution of finding the particle with a wave function at a given position and so it is the fundamental equation for modeling motions of atomic and subatomic particle systems, the solution of the CME is a probability distribution of the number of molecules in the time for molecular reactions systems [1]. Indeed, just as the Shrödinger's equation reduces to the Newton's second law assuming Planck's constant tends to zero, the Chemical Master Equation becomes the Law of Mass Action if the volume of reaction tends to infinity since we tend to the deterministic context.

3.3 One-Step processes reveal a Poisson distribution

The one-step or *birth-and-death* (BD) processes are a special class of Markov processes. Birth–death (BD) processes are characterized by two types of transitions: *births* which increase the state by one, and *deaths* which decrease the state by one. Many stochastic processes are BD processes, indeed they are suitable for modelling the dynamics of the number of individuals in a population, and are widely used in a broad range of areas such as biology, ecology and operations research. In particular, it can be exploited in order to model the evolution of a population of RNA molecules, since the population can increase (production) or decrease (degradation) by one molecule at a time [24].

More generally and rigorously defined, a BD process is a *continuous time Markov process* whose range consists of integers n and whose transition matrix $\mathbb{W}$ permits only jumps between adjacent sites [21]. In that case, the master equation (Eq 3.9) is written as

$$\frac{dP_n(t)}{dt} = W_{n,n+1}P_{n+1}(t) + W_{n,n-1}P_{n-1}(t) - W_{n-1,n}P_n(t) - W_{n+1,n}P_n(t) \tag{3.15}$$

The transition rates, $W_{n',n}$, are written in a special notation for these processes, i.e.

$$W_{n+1,n} = g_n \qquad\qquad W_{n-1,n} = r_n \tag{3.16}$$

Where g_n is the *gain* term, that is the probability per unit time for a jump from n to $n+1$ and r_n is the *recombination* term, that is the probability per unit time for a jump from state n to state $n-1$. The probability to jump to one more units in a time Δt is not impossible but is of the order of $O(\Delta t^2)$.

From here the meaning of the master equation as a gain-loss equation for the probabilities between states n is even more clear. It is similar to a balance of a fluid: some is entering and some is exiting.
Thus the master equation for one-step processes can be rewritten as

$$\frac{dP_n(t)}{dt} = \dot{P}_n = r_{n+1}P_{n+1} + g_{n-1}P_{n-1} - (r_n + g_n)P_n \tag{3.17}$$

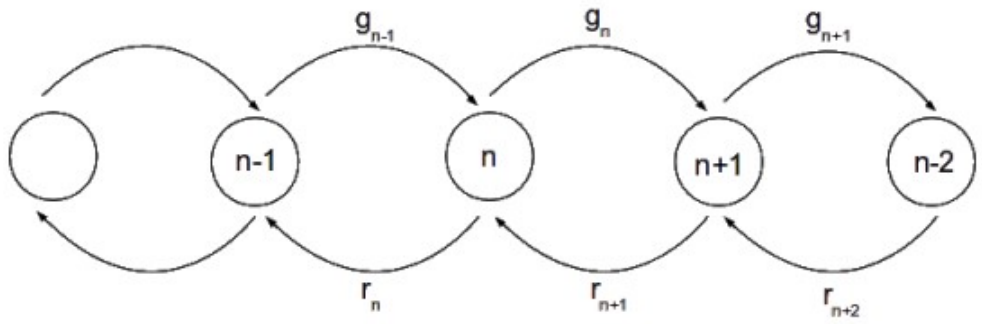

Figure 3.1: The one-step process with its transition probabilities and the different states. Image from [26].

However, we need to beware to the boundary conditions. Given that we have a number of molecules n that is $n \in [0, N]$, if $n = 0$ is a boundary, the Eq 3.17 is meaningless and has to be replaced with

$$\dot{P}_0 = r_1 P_1 - g_0 P_0 \tag{3.18}$$

since we need to set this condition $r_0 = g_{-1} = 0$ otherwise we would have a negative number of molecules which is physically unfeasible. Similarly an upper boundary $n = N$ requires a special equation:

$$\dot{P_N} = g_{N-1} P_{N-1} - r_N P_N \tag{3.19}$$

because the condition $r_{N+1} = g_N = 0$ or then with g_N we would have a transition from the state N to the state $N + 1$ and so to a state whose number of molecules is greater than the maximum number present in our system; a similar reasoning is done for r_{N+1} that means a transition from the state $N + 1$ to N.

One-step processes can be subdivided based on the coefficients r_n and g_n into the following categories: linear, if the coefficients are linear functions of n, nonlinear, if the coefficients are nonlinear functions of n and random walks, if the coefficients are constant [26, 21].
An important example of a one-step process with constant transition probabilities is the *Poisson process*, defined by

$$r_n = 0, g_n = q, p_n(0) = \delta_{n,0}, \tag{3.20}$$

where q is constant parameter. In such case, the master equation is

$$\dot{P}_n = q(P_{n-1} - P_n) \tag{3.21}$$

The solution of this equation is a time dependent probability distribution

$$P_n(t) = \frac{(qt)^n}{n!} e^{-qt} \tag{3.22}$$

that is a Poisson distribution. Let us keep in mind this theoretical result for the discussion about modeling gene expression in a stochastic framework in Chapter 4.

3.4 Resolution Methods for the CME

Solving the master equation would seem to be the ideal way of looking at stochastic systems, as it can tell us the full distribution of possible states the system can be in over time, but for one important disadvantage: its complexity.

In some special cases it is possible to solve it with analytical and numerical methods, for instance in case of linearity of the process where every reaction term is constant or proportional to the first power of only one chemical specie [5].

However, in general, what makes it difficult to solve is the mixing of a continuous time evolution with a discrete evolution of the state variables [49]. Furthermore, this becomes apparent when considering how many equations must be solved at each time point, and bearing in mind that the number of possible states can increase exponentially with time [12].

Thus we must use trajectory-based approaches, which differ from the master equation in that they do not find the distribution of all possible states over time. Rather, *they simulate single realizations*, meaning that at each step they choose one out of all the possible outcomes.

The trajectory of each stochastic simulation is different, since each is a single realisation sampled randomly from the full distribution given by the master equation. Given many of these random single trajectories, we can build up a picture of the full distribution.

Many approximation methods for the solution of the CME have been developed during the years and which operate by discretizing the time evolution. In this thesis, we will exploit the two simulation algorithms that have become the workhorse stochastic methods for many researchers today: the stochastic simulation algorithm (SSA) and the tau-leaping procedure. In the end, we will propose an "alternative" method that combines the two mentioned algorithms in order to speed up the stochastic simulation of a chemically reacting system.

3.4.1 Stochastic simulation algorithm (SSA)

Most of the attempts in solving the CME have concentrated on Monte Carlo simulations using the *stochastic simulation algorithm* (SSA) proposed by Gillespie in his seminal paper. This method is statistically exact, namely a full probability distribution built up from an infinite number of simulations will be identical to the distribution of the Markov process, given by the master equation [12]. This algorithm is arguably the simplest SSA variant and is based on the following steps in order to simulate *one stochastic realisation*:

assuming that the system is described by the state vector $\vec{Y}$ which in general represents the amount of molecules for each chemical species and given that the initial state of the system is $\vec{Y} = \vec{Y_0}$, one has to choose which reaction happens between the possible ones and how much time the system will spend in such state before the next reaction happens [5]. Each reaction i is described by its *transition rate* (or even called *propensity*) k_i and by the modification of the state vector that generate. Given that we are dealing with Markov process, the difference in time between two successive events Δt follows an *exponential distribution*, whose parameter is the propensity k_i:

$$p_i(\Delta t) = e^{-k_i \Delta t} \tag{3.23}$$

given that any reaction is independent from the others, the distance in time between two successive events is distributed as:

$$p(\Delta t) = e^{-\sum_{i=0}^{R} k_i \Delta t} \tag{3.24}$$

This process is iterated untill the whole time of interest has been simulated or untill an other desired condition is achieved.

Unfortunately, the direct method SSA pays the price for its simplicity in computational time, as two random numbers must be generated at each step [12] (the one for choice of the event and the other for the choice of the time spent in the state given by an exponential random variable). Furthermore, the SSA has an inherent limitation: it must simulate every single reaction. This is, of course, its great strength too, but in cases where there are many reactions or larger molecular populations, it often becomes too slow to generate useful numbers of simulations in a practical time period.

3.4.2 Tau-leap algorithm

The tau-leap method, or also referred as explicit tau-leaping procedure, is similar to the SSA in that each simulation is a *single stochastic realisation* from the full distribution at each step [12]. However, the steps are fixed and in each one of these *it counts the total number of occurrences of each type of reaction over that step approximating the number of firings of each reaction channel during a chosen time increment τ as a Poisson random variable*. Thus, also in this case based on the initial state of molecular populations $\vec{Y} = \vec{Y_0}$, if there are M reactions, we take M Poisson random number samples. We can call the time interval in which these reactions happen as τ. Hence, the vector of molecules given by the M reactions $\vec{Y} = \vec{Y}_M$ in the time interval τ is given by

$$\vec{Y}_m(\tau) = P(\vec{K}_i \tau) \tag{3.25}$$

where P is the Poisson distribution with mean equal to $\vec{K}_i \tau$ of which $\vec{K}_i$ is the vector of propensities. Then the initial state of molecular populations is updated based on Eq

3.25 and this is iterated untill the whole time of interest has been simulated or a desired condition is met, similarly to SSA.

Hence, the basic idea of this procedure is to advance the system by a *pre-selected* time increment τ (in contrast to the generated time increment that is in the SSA), which is large enough that many reaction events occur in that time, but nevertheless small enough that *no propensity function value is likely to change "significantly" as a consequence of those reaction events* [13].

Summing all up, tau-leap algorithm has the key advantage that we now do not need to spend time simulating individual reactions and as such it is much faster than the SSA. On the other hand its main drawback is that we have lost accuracy compared to the SSA in some ways such as we do not know when each reaction occurred within the time step.

The accuracy issue can be mitigated (at the expense of computational time) as the error level, and so time step size, is decreased; indeed, as the time step tends to zero, the tau-leap effectively becomes the SSA, with each step experiencing either zero or one reaction [12].

Moreover, as anticipated in some rows above, there is another complementary issue with the tau-leap: as many reactions are simulated at once, molecular species that are depleted in any reaction can go negative if the time step is too large. This is obviously unphysical, and so undesirable and makes it inappropriate to simulate reactions that regards species which are typically lower in number such as genes.

Fig. 3.2 shows the number of RNA molecules and proteins produced during time obtained by using the just explained stochastic simulation methods (SSA and Tau-leap) and those given by the deterministic approach, i.e. ordinary differential equation solution, considering a model with a gene that is always expressed (see section 2.1). In this particular case in which the RNA production rate is higher than the degradation rate and the same is for proteins, we do not run the risk to obtain unphysical negative number of molecules and we can compare the results given by the algorithms. In particular, we consider the following values of rates: gene activation $k_a = 1$, gene deactivation $k_i = 0$, RNA production $k_1 = 1$, RNA degradation $k_2 = 0.1$, protein production $k_3 = 1$, protein degradation $k_4 = 0.1$.

From Fig. 3.2, we notice that trends resulting from the three methods are similar both for the number of RNA molecules and for the number of proteins produced during time.

However, in order to compare the two stochastic simulation methods, it is more useful to compare the stationary distributions (i.e. the distribution of states) obtained instead of the time dependent results.

Hence, we run the same simulation till a longer time limit to allow to obtain the expected Poisson distributions. The results are shown in Fig. 3.3:

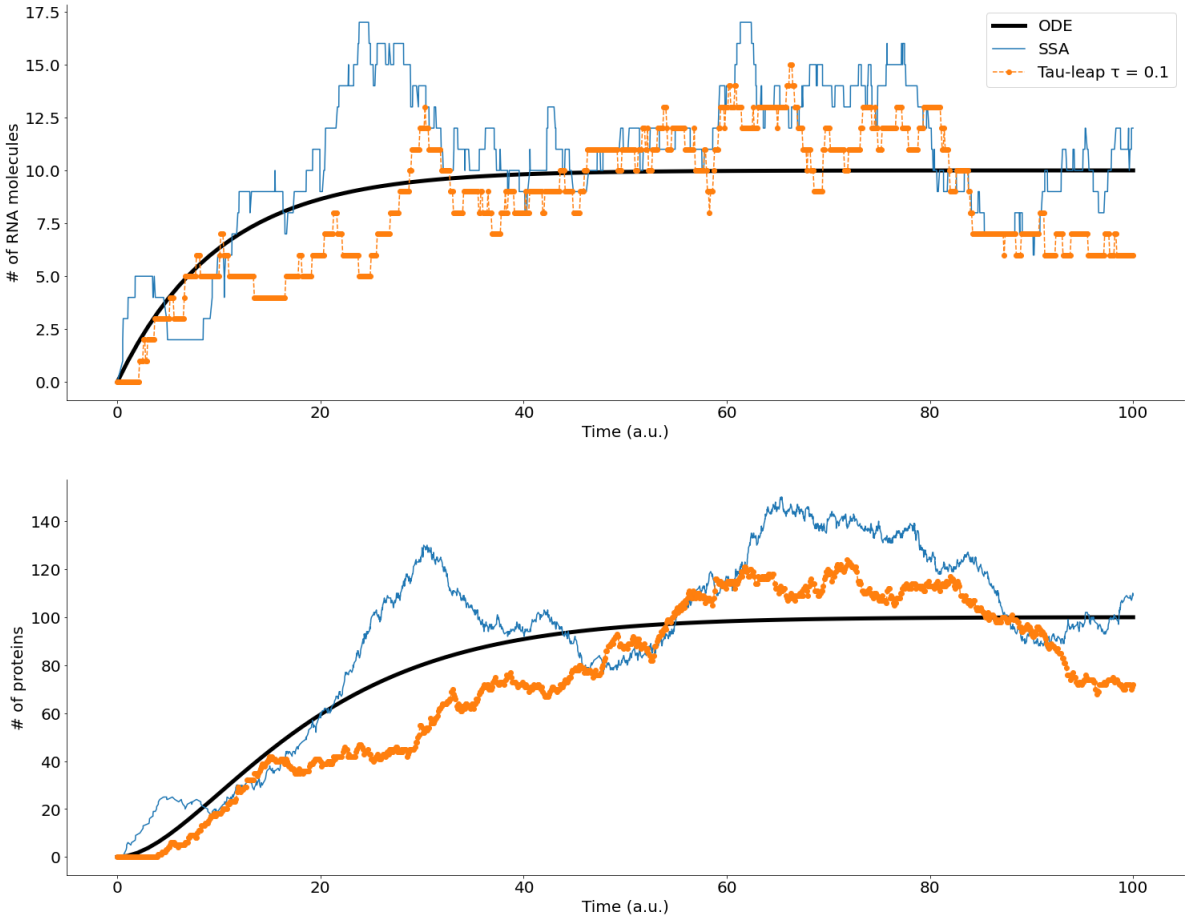

Figure 3.2: Comparison of simulation methods. The typical trajectory of the number of molecules as function of time is reported from the SSA (blue line) and tau-leap (orange dotted line) and ODE solution (black thick line). The upper plot refers to the number of RNAs molecules produced as a function of time starting from a gene that is always active, while the plot below represents the number of proteins produced starting from the RNAs of the same model.

There is a clear overlap between the two distributions, showing that the SSA simulation is providing the same results to the Tauleaping, validating also that the two algorithms work in representing the stochastic trajectory of the chemical reacting system we are analysing.

The time plot referred to the stationary distribution shown in Fig. 3.3 is the one reported in Fig. 3.4:

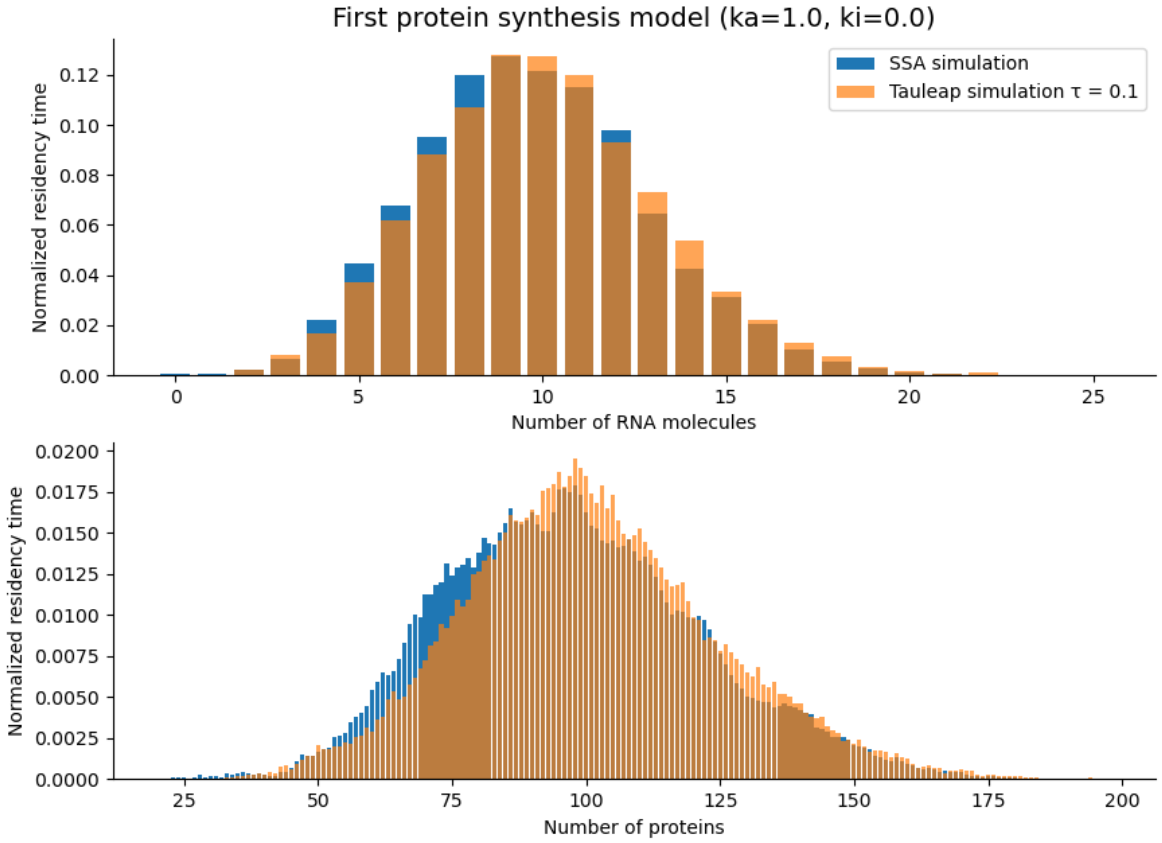

Figure 3.3: Comparison of simulation methods by using the stationary distributions given by the SSA (orange bar plot) and tau-leap (blue bar plot).

Also in Fig. 3.4, the ODE solution is reported. We can notice that the deterministic approach, even though cannot be considered representative of the true behaviour of the system, its result represent a useful statistic of the system that in this case is the means of our unimodal symmetric distributions of the number of RNAs and proteins produced. The black line in Fig. 3.2 and 3.4 is in fact a rolling mean of the stochastic simulation.

Furthermore, deterministic simulations are generally much faster to run than their statistical counterparts [12].

In order to make this simulation for a virtual time equal to 9000 with a computer of 1.10 GHz CPU, the SSA simulation has spent 55 seconds with respect to the 10 seconds spent by the explicit tau-leaping algorithm.

However the tau-leaping algorithm can work well in this situation of constitutive gene

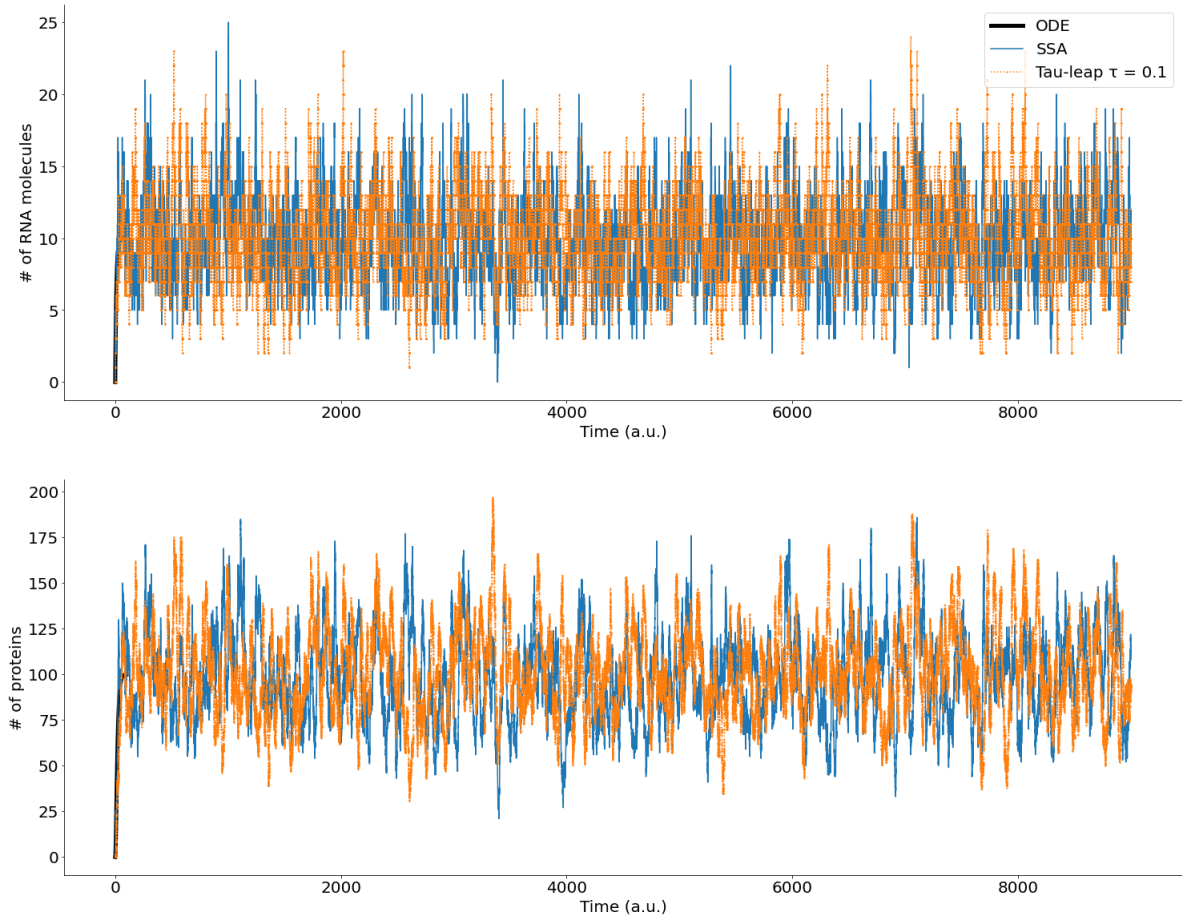

Figure 3.4: Comparison of simulation methods using a higher time limit than the one in Fig.3.2. The typical trajectory of the number of molecules as function of time is reported from the SSA (blue line) and tau-leap (orange dotted line) and ODE solution (black thick line).

expression. And what if our gene is regulated ? The risk of running in negative number of molecules is much higher hence we need an algorithm that checks that this does not occur if we want to go on using the tau-leap procedure.

In this kind of system, the typical choice is to combine the SSA and the tau-leaping procedure because the first one can simulate the gene behavior while the second one is suitable for simulating the changing of more numerous molecular species such as RNAs and proteins.

In the next section, we propose and describe a combined SSA/tau-leaping algorithm.

3.4.3 An alternative method: the "hybrid"stochastic simulation algorithm

Many schemes have been developed that allow the choice of larger time steps in the Tau-leap algorithm whilst avoiding negative populations. In this thesis work, we have tried an approach that combines, basically, the SSA with Tau-leap algorithm in a way that the first one updates the state of genes while the latter updates the state of other more numerous species such as RNA molecules and proteins.

The model that we consider to develop this "hybrid" algorithm is the first simple protein synthesis model (the one described in the first section of Chapter 2). We have 3 molecular species: one *gene* whose transcription leads to *RNAs* production that are translated into *proteins*. In other words, this is what is called as *the central dogma of biology*, that states that genes specify the sequence of mRNA molecules, which in turn specify the sequence of proteins (DNA $\rightarrow$ RNA $\rightarrow$ Protein). Hence we have 6 possible reactions: gene activation, gene deactivation, RNA production, RNA degradation, protein production and protein degradation.

Let us consider our usual state of molecular populations $\vec{Y}$ and the initial state $\vec{Y} = \vec{Y_0}$. The transition rates related to such state is $\vec{K} = \vec{K_0}$.

The proposed hybrid algorithm is based on the following steps:

1. In state $\vec{Y}$ at time t, evaluate the transitions rates of all species and those related to genes $\vec{K}_{genes}$.

2. Apply the SSA algorithm by considering the sum of genes transition rates in Eq.3.24 and hence select the reactions that modify the time of residency of the gene state.

3. Eq.3.24 represents in this case the time of residency of the gene $t_{gene\_state\_residency}$ in the initial state. The aim of this algorithm is to speed up the SSA by considering such $t_{gene\_state\_residency}$ as the tau τ in the Tauleaping.

4. Thus, after the SSA application, consider the $t_{gene\_state\_residency}$ and the updated gene state $\vec{Y}'$.

5. Apply the Tauleaping algorithm to the RNAs and proteins state and by considering $t_{gene\_state\_residency}$ as τ, so $t_{gene\_state\_residency} = \tau$.

6. Consider the updated state by the tauleaping and SSA $\vec{Y}'$ and compute also the difference between the transition rates $\vec{K_0}$ refered to the initial state $\vec{Y_0}$ and the final value of transition rate $\vec{K}'$ obtained considering the updated state $\vec{Y}'$. In particular compute the following difference:

$$\Delta\vec{K} = \left| \frac{\vec{K}'}{K'_{TOT}} - \frac{\vec{K}}{K_{TOT}} \right| \tag{3.26}$$

where K'_{TOT} is the sum of all the transition rates calculated with the new updated state $\vec{Y}'$ and the same for K_{TOT} that is the sum of all the transition rates calculated with the initial state $\vec{Y}_0$.

Let us rember indeed that we are dealing with first order kinetics reactions, whose transition rates are propotional to the amount of molecules of interest by a constant k.

$\vec{Y}'$ and $\Delta\vec{K}$ are the two quantities that must be monitored, because, as regards the first one, of course, we need to check that there are not negative number of molecules; for the latter, if it is too large, it can lead in very low accuracy in the algorithm results. For instance, if there is a tauleaping step that leads to a final state $\vec{Y}'$ that has a much higher number of proteins with respect to the initial state $\vec{Y}_0$, the final stationary distribution will result in an "explosion" in the number of molecules that does not represent the true distribution.

Thus, in the following steps we describe how we monitor these two quantities.

7. If $\vec{Y}'$ has all positive values and $\Delta\vec{K}$ is lower than a given threashold chosen by the user, update rates and the state by considering the results of the SSA and tauleaping algorithms already applied in points 2 and 5. If the simulation is running untill a precise time limit (as we always do in this thesis work), update the time of observation using the $t_{gene\_state\_residency}$.

8. If instead $\vec{Y}'$ has a negative number of molecules or one of $\Delta\vec{K}$ is higher than the threashold, it means that we need to make smaller steps than the one proposed by the SSA.

   Hence we divide by one half τ, $\tau' \to \tau/2$, untill the conditions on the number of molecules and on the $\Delta\vec{K}$ are satisfied.

   When we obtain the right τ', we use it to make a tauleaping, updating the RNAs and proteins states.

   Then, we attempt a tauleaping using the remaining time $\tau - \tau'$ that we call as τ''.

   If the conditions are satisfied, we update the total time of observation with τ and attempt a new tau-leaping starting from point 1 by considering a new updated state $\vec{Y}'$ and the new rates $\vec{K}'$.

   If it is not, we divide again τ'' by one half, $\tau''' \to \tau''/2$ untill the desired conditions are met.

   This process is iterated untill the $\tau^{(n)}$ is higher than 3 times the characteristic time t_c given by the SSA by considering all reactions happening in our system.

9. If the new $\tau^{(n)}$ reaches a lower value, i.e. $\tau^{(n)} \leq 3t_c$, it is worth to make a SSA step without getting trapped in the previous iterating system that is certainly less

precise than the SSA. In this case, update the total time with the characteristic time given by the SSA applied considering all the reactions that are possible in our system.

10. Apply steps from 1 to 9 untill the desired total time is reached or the desired condition is met.

As in the tauleaping section, we have compared the results of the new algorithm stationary distribution with the SSA. We consider the following model parameters: activation rate $k_a = 1$, deactivation rate $k_i = 0.5$, RNA production rate $k_1 = 1$, RNA degradation rate $k_2 = 0.1$, protein production rate $k_3 = 1$, protein degradation rate $k_4 = 0.1$. These are the same parameters and the same model used to compare the SSA simulation with the tauleaping simulation except for the fact that the deactivation rate is now different from 0. Indeed now we can simulate without worrying about the fact that the number of molecules becomes negative and so we can add gene regulation to our model.

We consider a threashold $\Delta K_t = 10$ and the result for our basic model is promising.

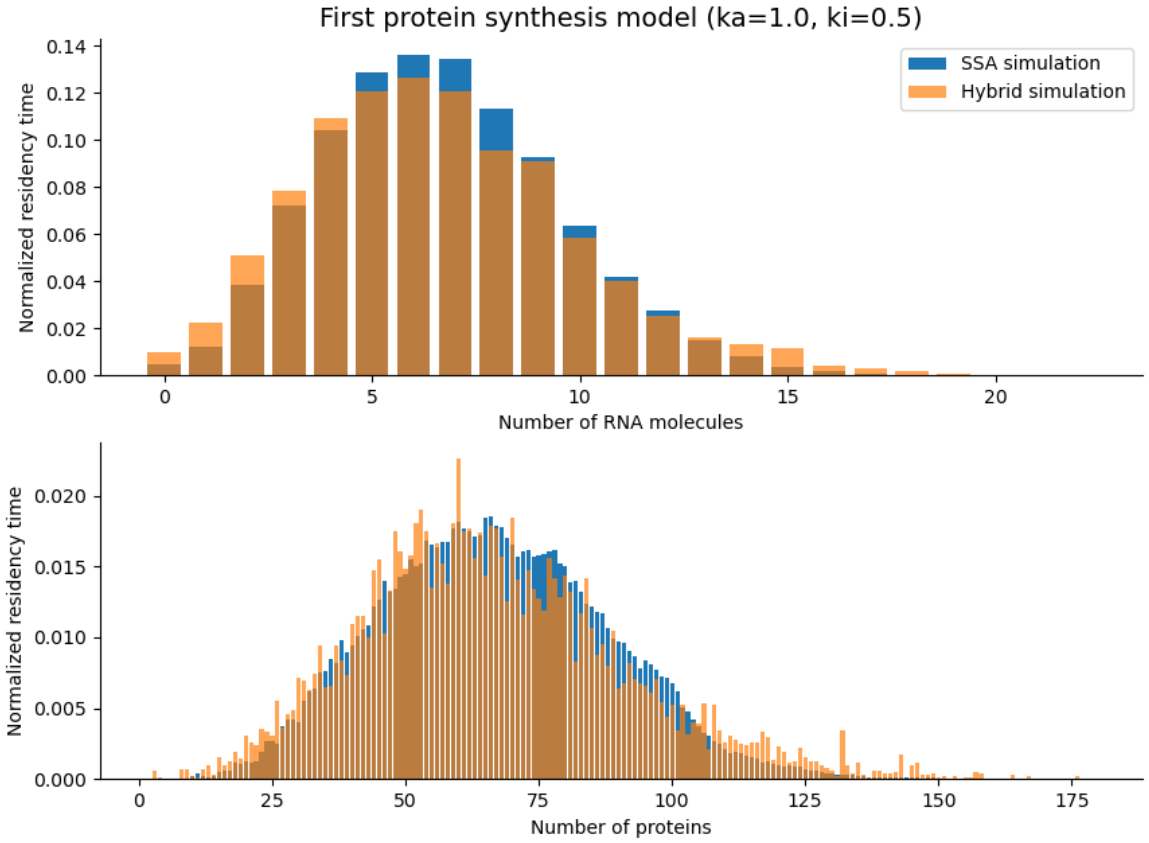

Figure 3.5: Comparison of simulation methods using stationary distributions. The blue bar plot is the result given by the SSA algorithm while the orange bar plot is the result given by the hybrid algorithm.

Fig. 3.5 shows the distribution of states given by the SSA (blue bar plot) and the hybrid algorithm (orange bar plot). They show a high degree of overlapping with hybrid approach speed that is about 10 times lower than the SSA. More precisely, using a computer with 1.10 GHz, the speed results are summed up in the table 3.1.

	speed
SSA	107s
Hybrid	7.5s

Table 3.1: Computational time required for the SSA and Hybrid algorithm in case of our basic model.

We can modify the model changing the parameter value k_1 (that in the end it is also at the basis of the distribution). By considering $k_1 = 10$, the related stationary distribution is shown in Fig. 3.6.

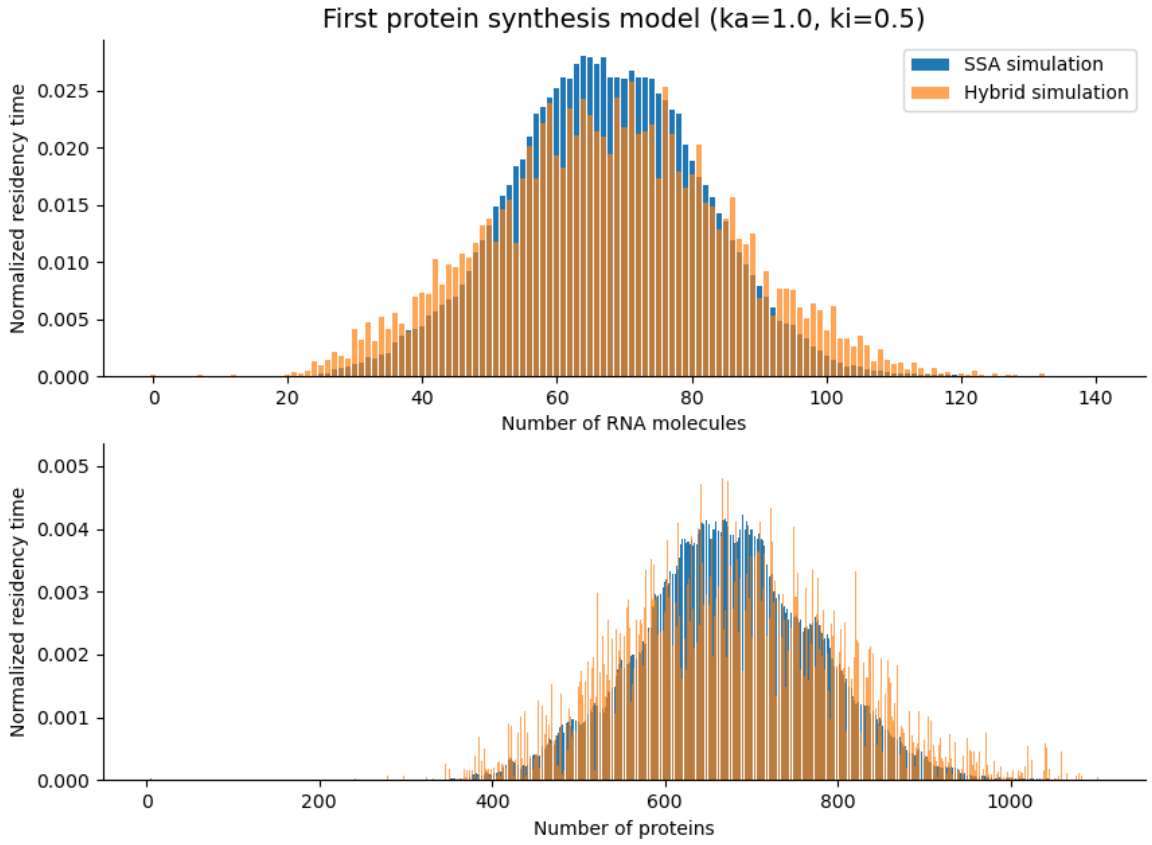

Figure 3.6: Comparison of simulation methods using stationary distributions with $k_1 = 10$. The blue bar plot is the result given by the SSA algorithm while the orange bar plot is the result given by the hybrid algorithm.

There is still a high degree of overlap between the two distributions even though the protein hybrid distribution result is more scattered with respect to the SSA. In this case, the gain in velocity is really high. Once again, the relative speeds are reported in the table 3.2.

	speed
SSA	718s
Hybrid	8s

Table 3.2: Computational time required for the SSA and Hybrid algorithm is case $k_1 = 10$.

However, the situation changes if we decrease 10 times the rate of RNA production and so $k1 = 0.1$ instead of increasing it by 10 times as in the previous case.

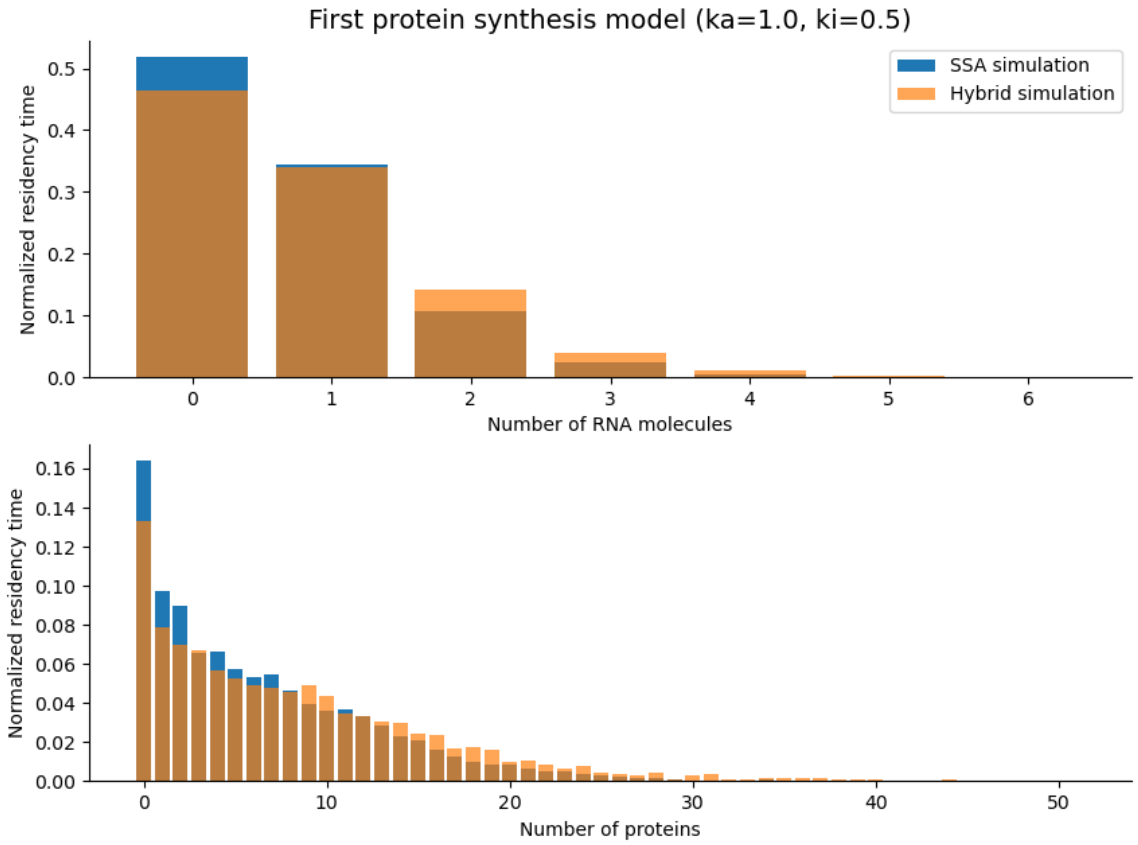

Figure 3.7: Comparison of simulation methods using stationary distributions with $k_1 = 0.1$. The blue bar plot is the result given by the SSA algorithm while the orange bar plot is the result given by the hybrid algorithm.

The degree of overlap is still high, however we do not achieve a considerable improvement in velocity:

	speed
SSA	8.3s
Hybrid	7.9s

Table 3.3: Computational time required for the SSA and Hybrid algorithm is case $k_1 = 0.1$.

This suggests that decreasing the number of RNAs produced, the performance of our hybrid algorithm decreases. What if we consider the opposite type of regulation ?

Let us consider $k_a = 0.1$ and $k_i = 1$ and let the other parameters equal to the basic model values.

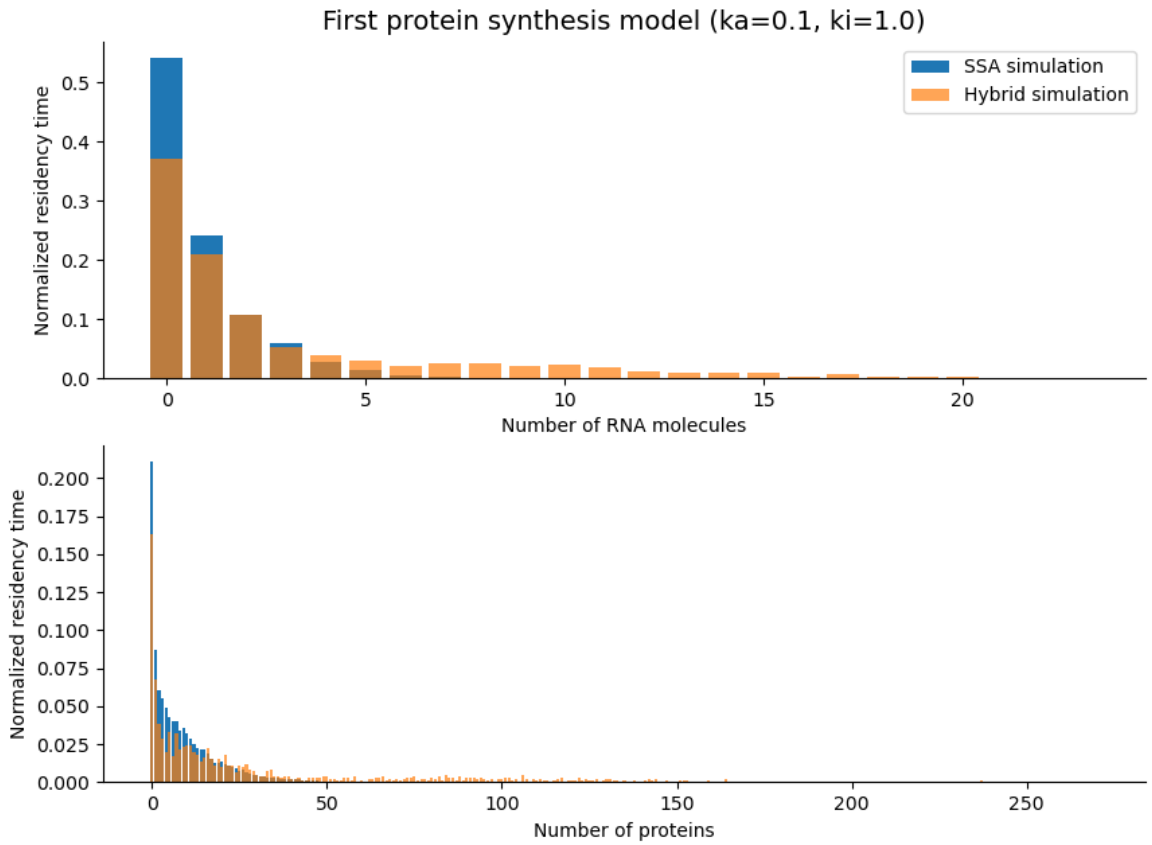

Figure 3.8: Comparison of simulation methods using stationary distributions considering gene activation rate lower than the deactivation rate ($k_a = 0.1$ and $k_i = 1$). The blue bar plot is the result given by the SSA algorithm while the orange bar plot is the result given by the hybrid algorithm.

Fig. 3.8 shows the effects of this other kind of regulation on the distribution given by the two algorithms. As expected, in both cases the results show that the residency time refered to the state with the number of molecules equal to zero is the highest. However, in this case, the hybrid algorithm shows an increase in the number of molecules state both in the RNAs and even more in the proteins states. One can try to decrease the

	speed
SSA	9s
Hybrid	35s

Table 3.4: Computational time required for the SSA and Hybrid algorithm is case $k_a = 0.1$ and $k_i = 1$.

threashold ΔK_t but at expence of computational time that is already 3 times higher than the SSA (see table 3.4).

Hence we can conclude that this proposed hybrid algorithm works very well from the point of view of accuracy and computational time when the model tends to produce a high number of molecules, whereas we still do not find an approach that is better than Gillespie's algorithm when the number of molecules produced is low. Anyway, in such cases of low number of molecules, the SSA is efficient and there is not a huge needing to be optimized.

Furthemore, we need to notice that our algorithm is a "paramater-dependent" algorithm, where the parameter is Δk_t. As regards how tuning this parameter, we can suggest to keep its value as high as possible if one wants to increase the speed but paying attention to the accuracy because with an higher Δk_t you will run the risk of "the explosion of molecules".

One can notice, indeed, that the logic of our hybrid algorithm is similar to the Adaptive Runge-Kutta methods used for the approximate solutions of simultaneous nonlinear equations: during the integration, the step size is adapted such that the estimated error stays below a user-defined threshold. If the error is too high, a step is repeated with a lower step size; if the error is much smaller, the step size is increased to save time [61].

Anyway, in this thesis work we will use the Stochastic Simulation Algorithm because our hybrid approach needs to be further tested and improved.

3.5 Statistical analysis techniques for warm-up detection

Up to now we have shown the results of stationary distributions of the CME for comparing different algorithms. However we have omitted to explain an important analysis step when dealing with simulations: the warm-up detection and its removal.

We can define our simulations as "infinite horizon simulations": it has no natural beginning point or ending point. The desired starting conditions are sometimes in the far off future. We are interested in the long-run performance of the system and since there is no natural ending point of interest, we consider the horizon infinite [60].

As mentioned in section 5.1, sometimes we are interested in finding the moments of the probability distributions, such as the mean.

It is true that if the steady state distribution exists and you run the simulation long enough the estimators will tend to converge to the desired quantities. You can observe it in plots reported in Fig. 3.2 and 3.3. At the beginning the number of molecules is low. Then by increasing virtual time, they increase untill they reach a state that we can call as "steady" state, that means that the distribution of the desired performance measure has reached a point where it is sufficiently similar to the desired steady state distribution [60].

We have also mentioned in section 3.2, when talking about Chemical Master Equation solutions, that the eigenvector called also as the metastable state is the longest-standing one and its eigenvalue gives a time-scale of the time of convergence to the stationary distribution.

We need to clarify that "steady state" is a concept involving the performance measures generated by the system as time goes to infinity. It does not mean that the system itself has reached steady state.

The system never reaches steady state. If the system itself reached steady state, then by implication it would never change with respect to time. The system continues to evolve with respect to time.

The period before the so called steady state can affect the mean calculation and so also the distribution of states. This period is called warmup-period.

At the end of the warm up period, the system can be in any of the possible states of the system. Some states will be more likely than others.

Thus, within the infinite horizon simulation context, you must decide on how long to run the simulations and how to handle the effect of *the initial conditions* on the estimates of performance by estimating when the warmup period ends, i.e. a time point that we can indicate also with T_w.

The initial conditions of a simulation represent the state of the system when the simulation is started.

If T_w is long enough, then on average across the replications, you are more likely to start collecting data when the system in states that are more representative over long term (rather than just states with low number of molecules).

3.5.1 The effect of initial conditions

Consider the output stochastic process Y_i of the simulation and let $F_i(y|I)$ be the conditional cumulative distribution function of Y_i where I represents the initial conditions used to start the simulation at time 0. If $F_i(y|I) \to F(y)$ when $i \to \infty$ for all conditions I, then $F(y)$ is called the steady state distribution of the output process.

The distribution $F_i(y|I)$ at the start of the simulation, however, tends to depend more heavily upon the initial conditions. Estimators of steady state performance, such

as the sample average, will tend to be biased.

If the expected value of the sampling distribution is equal to the parameter of interest then the estimator is said to be unbiased, i.e. $E[\hat{\theta}] = \theta$, with $\hat{\theta}$ a point estimator.

If instead the estimator is biased then the difference $E[\hat{\theta}] - \theta \neq 0$ and this difference is the bias of the estimator $\hat{\theta}$.

This is the so called initialization bias problem in steady state simulation or warm up problem. The warm-up period d is the period such that given Y_i with $i = d+1, ..., Y_i$ *will have substantially similar distributional properties as the steady state distribution.*

Unless the initial conditions of the simulation can be generated according to $F(y)$, which is not known, you must focus on methods that detect and/or mitigate the presence of initialization bias.

3.5.2 "Replication-deletion" method

There are many methods that aim to determine the warmup period. In this thesis work we have used the a strategy that we can call "replication-deletion" method because it is based on the following steps:

1. Make R replications. Typically $R \geq 5$ is recommended.

2. Since the sample points in SSA are obtained at different time intervals for each simulation, we interpolate them at constant intervals (we choose $dt = 0.01$).

3. Thank to the previous step, we can make the average across the replications.

   Let Y_{rj} be the j^{th} observation on replication r $j = 1, 2, ..., m$ where m is the minimum of the number of observations in the r^{th} replication, $m = min(m_r)$.

$$\hat{Y} = \sum_{r=1}^{n} Y_{rj} \tag{3.27}$$

4. Find T_w in $\hat{Y}$ by calculating the difference between $\hat{Y}$ at t and at t+dt, t+dt and t+2dt etc... when the differences start to become constant, then t is T_w that we are looking for.

Fig. 3.9 show the concept of a warm up period for a simulation replication.

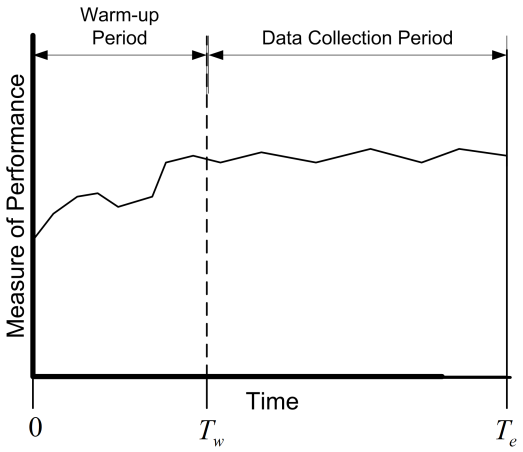

Figure 3.9: Example of a simulation with warmup period detection. Image from [60].

3.6 Absorbing states

From a computational standpoint, we could decide to allow an infinite horizon simulation become finite by considering an ending condition that is the absorbing state, that is a state for which all transition rates are zero. This means that there are no transitions that can lead out of that state and, as a consequence, the residency time is infinite.

In our biological context this situation can not happen because if we consider that RNAs molecules are transcripted from a gene and then, from those, proteins and translated and produced, the gene can be in active or inactive form but a gene can not degrade. Hence we will always have at least one molecule and this condition can not happen. In our simulations, however, when comparing the different algorithms, we have considered it as possibile with a rate much lower with respect to the others, for a computational convenience. It is like an additional test that one can do in order to assess that the algorithm works and do not tend to degrade its molecules.

Part II

Studying genetic circuits and noise

Chapter 4

Genetic circuits simulations in a cellular environment

In chapter 2 we described in a deterministic framework the three simple genetic circuits at the basis of living organisms. Now that we have deepened into stachastics simulations, we can look at their time course behavior using the SSA algorithm in a stochastic framework that is the one that we typically encounter in a cellular environment, as explained in the introduction. Additionally to this, we provide information about their "place" in biology in order to have an idea of what we are simulating and then investigating in Chapter 5.

4.1 Constitutive expression and Poisson statistics

Typically, constitutive genes are the so called "housekeeping genes" that are required for the maintenance of basic cellular function [50], for instance genes coding for ribosomal proteins.

For example, Zenklusen et al. in the experiments [51] found that constitutive gene expression model offers surprisingly good quantitative matches to transcriptional behaviors for the housekeeping genes MDN1, KAP104, and DOA1 in budding yeast. The measured numbers of mRNA transcripts per cell were well described by Poisson distributions for all three genes.

As mentioned in section 3.3 when discussing about one-step processes, from a mathematical point of view, constitutive expression can be seen as a particular case of a one-step process since such molecular reactions are random Poissonian birth-death processes. This is represented in Fig.4.1 that reminds to Fig.3.1 of the previous chapter.

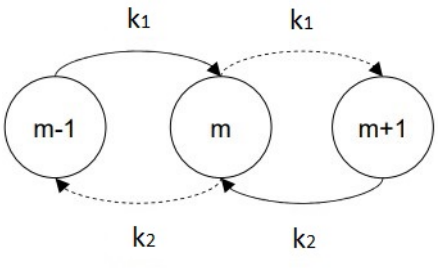

Figure 4.1: For unregulated mRNA synthesis, the system can move into the state of interest m in two ways, indicated by solid arrows. The system can likewise move out of the state of interest, indicated by dashed arrows.

Thus in such case, the distribution of RNA molecules produced has a Poissonian shape.

This is not the case of proteins distribution. The number of proteins produced yields a *scaling law* since they are dependent on both mRNA birth and death as well as translation rates [12]. Therefore proteins have non-Poisson distributions albeit their scaling is also roughly Poissonian.

Fig. 4.2 shows the time course trend of RNA molecoles and proteins produced by considering this model of gene expression by using the SSA.

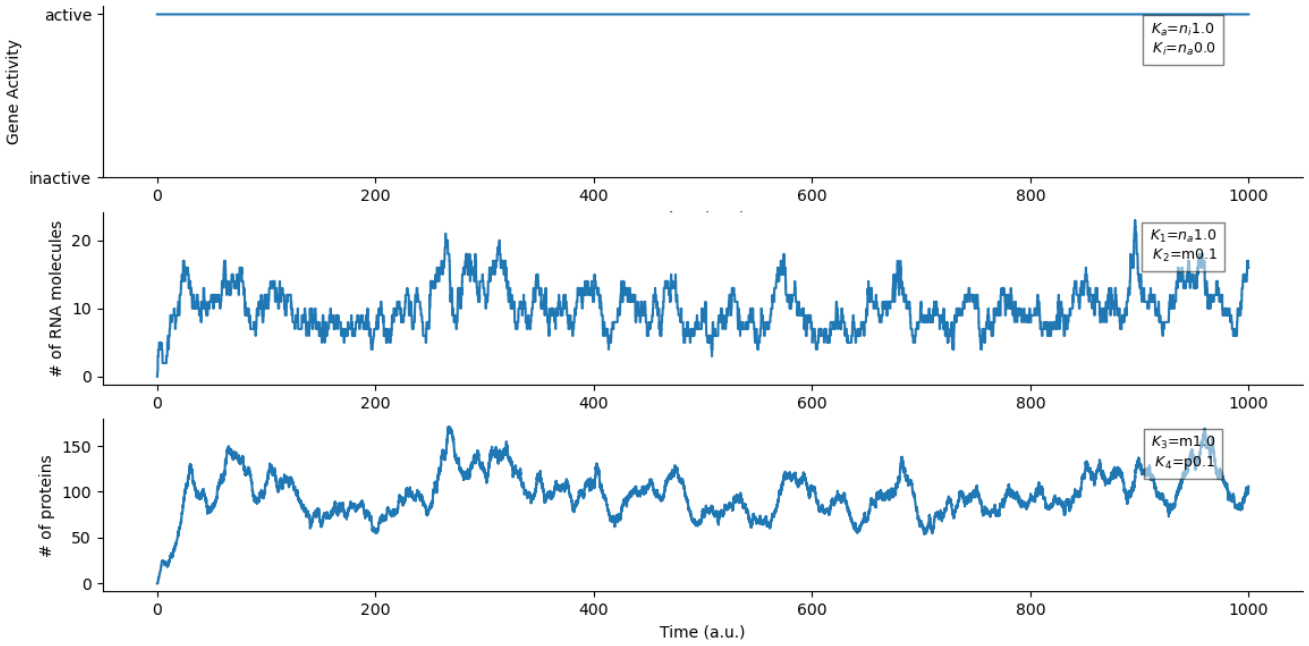

Figure 4.2: SSA time course simulation of the number of RNAs and proteins molecules from a gene (of the model in Fig.2.2) that is always expressed. The first plot shows the gene activity, the second one the number of RNA molecoles produced and the third one shows the number of proteins produced. Transition rates values are reported in image boxes.

We can notice the warm-up period at the beginning of the simulation discussed in section 3.5.

Moreover, we can observe the stiffness of the process: the number of proteins reacheas the steady state in correspondence to a higher number of molecules with respect the RNAs in the same or very similar warmup period (if we look at the time scale). Proteins indeed are produced faster than RNA molecules. This is reasonable if we think about the fact that we are dealing with first order kinetics systems.

4.2 Regulated gene expression

Although the constitutive gene expression model captures the fluctuations of several housekeeping genes, it does not perform as well when gene expression is regulated.

Deviations from Poisson behavior indicate regulation.

Fig. 4.3 and Fig. 4.4 show the number of RNA and molecules produced, using the SSA simulation, by considering respectively two types of regulation: one when the gene spents more time to be in active state than inative state whereas the other is the opposite situation.

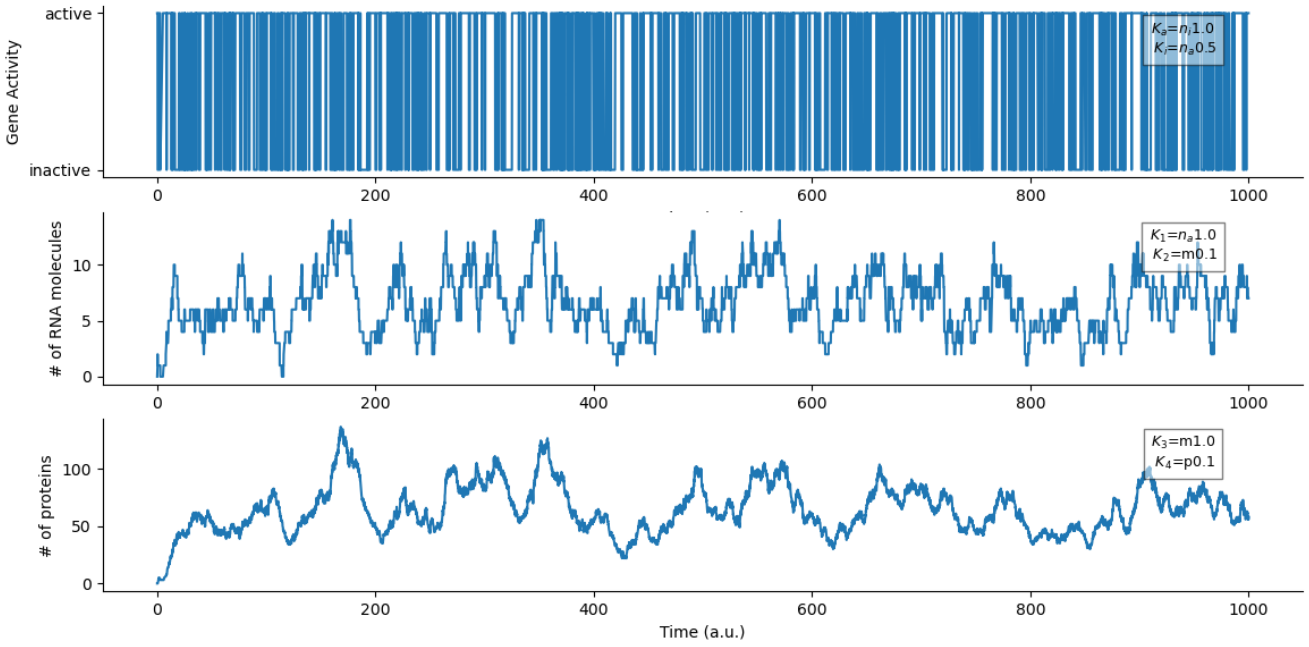

Figure 4.3: SSA time course simulation of the number of RNAs and proteins molecules from a gene (of the model in Fig.2.3) that is regulated with activation rate constant $k_a = 1$ and $k_i = 0.5$. The first plot shows the gene activity, the second one the number of RNA molecoles produced and the third one shows the number of proteins produced. Transition rates values are reported in image boxes.

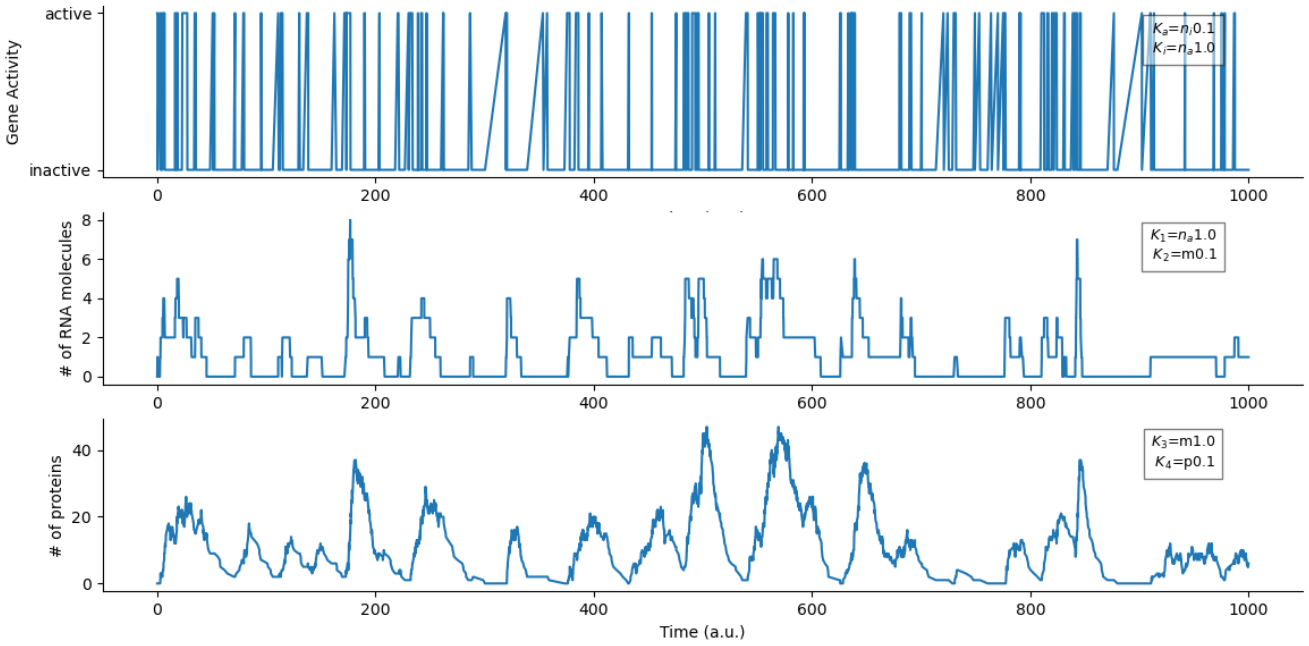

Figure 4.4: SSA time course simulation of the number of RNAs and proteins molecules from a gene (of the model in Fig.2.3) that is regulated with activation rate constant $k_a = 0.1$ and $k_i = 1$. The first plot shows the gene activity, the second one the number of RNA molecoles produced and the third one shows the number of proteins produced. Transition rates values are reported in image boxes.

The other two cases that we are going to simulate are the autorepressor case and the toggle-switch. We can anticipate how the pattern of molecules produced related to the case of the gene that spents more time to be inactive (see Fig. 4.4) is similar to the autorepressor case that we are going to explore.

4.3 The autorepressor simulation

Fig.4.5 shows the typical pattern of the number of transcript and translated molecules produced in a genetic autorepressor circuit.

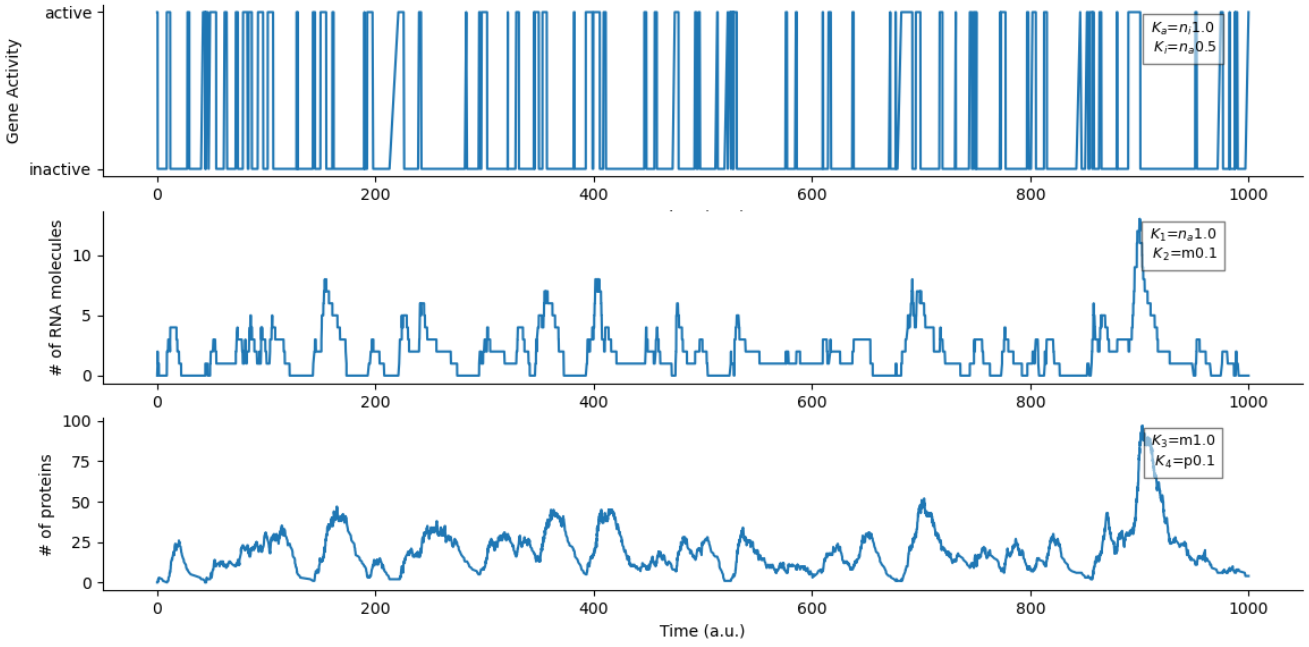

Figure 4.5: SSA time course simulation of an autorepressor model. The first plot shows the gene activity, the second one the number of RNA molecoles produced and the third one shows the number of proteins produced. Transition rates values are reported in image boxes.

The typical "sawtooth" pattern that is repeated over time is revealed by the SSA simulation in RNA molecules trend. As regards proteins time trend, the scaling law mentioned in section 2.2 causes more "smooth" edges.

One interesting example of autorepressor model in humans is the case of Regulatory Factor X_1 gene (RFX1).

This gene encodes a member of the regulatory factor X (RFX) family of transcription factors, which are characterized by a winged-helix DNA-binding domain. The encoded transcription factor contains an N-terminal activation domain and a C-terminal repression domain, and may activate or repress target gene expression depending on cellular context. This transcription factor has been shown to regulate a wide variety of genes involved in immunity and cancer, including the MHC class II genes and genes that may be involved in cancer progression. This gene exhibits altered expression in glioblastoma and the autoimmune disease systemic lupus erythematosis (SLE) [53].

Regulatory factor X1 (RFX1) is an evolutionary conserved transcriptional factor that influences a wide range of cellular processes such as cell cycle, cell proliferation, differentiation, and apoptosis, by regulating a number of target genes that are involved in such processes. On a closer look, these target genes also play a key role in tumorigenesis and associated events. Research work compile significant evidence for the tumor-suppressive activities of RFX1 while also analyzing its oncogenic potential in some cancers [54].

One of the most understood ways of regulating RFX1's gene expression is by its autorepression. RFX1 auto-represses itself with the assistance of yet unidentified co-repressor, in response to DNA replication arrest, to bring about a controlled expression of RFX1 [54]. Such gene is functionally conserved for instance in yeast where we find a yeast RFX1 homologue, that is Ctr1. There are evidences for a common mechanism for Crt1 and Rfx1 expression and for the conservation of their mode of action in response to a DNA replication block [55].

4.4 The Toggle-Switch simulation

Fig.4.6 shows the typical trend of gene expression in a toggle switch system.

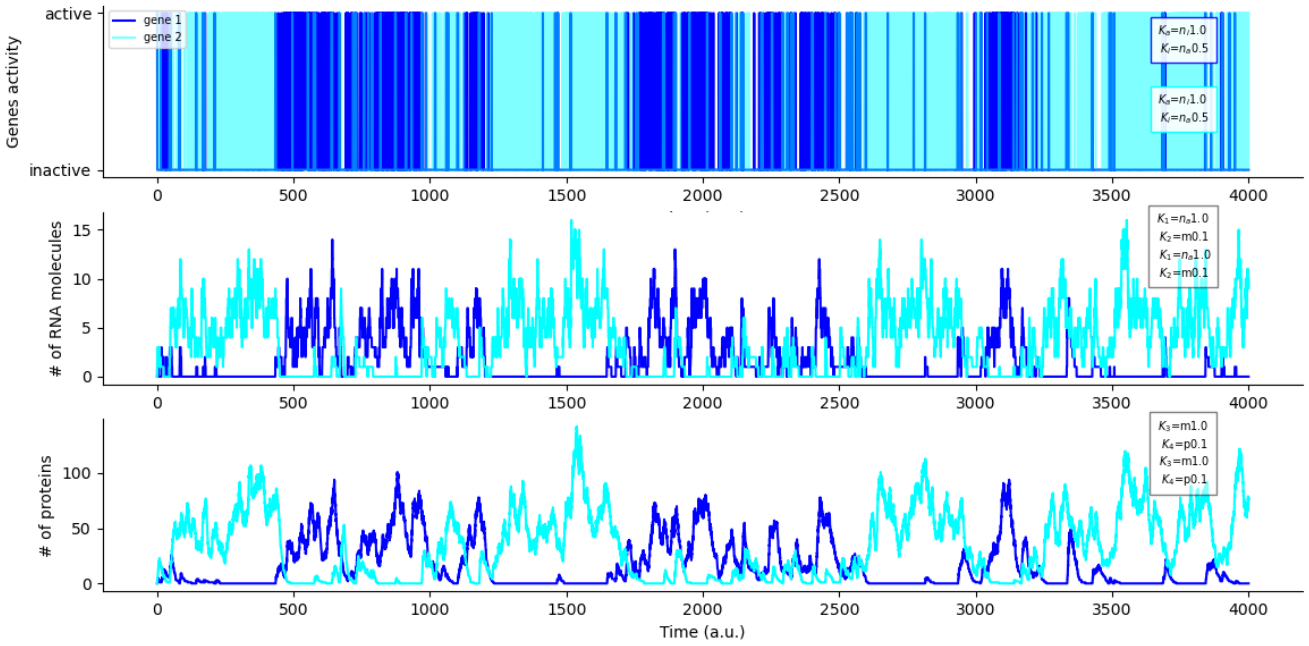

Figure 4.6: SSA time course simulation of a toggle switch model. The first plot shows the gene activity, the second one the number of RNA molecoles produced and the third one shows the number of proteins produced. Transition rates values are reported in image boxes.

We can clearly see the alternation between the expression of the two genes.

One example of a genetic toggle switch is the so called MicroRNAs Toggle Switch analysed in the work *Stochastic analysis of a miRNA-protein toggle switch* [56] by Giampieri et al.

MiRNAs are small, non-coding RNAs that modulate the expression of target mRNAs. MiRNAs are often part of toggle switches, with important examples are gene pairs built with oncogenes and tumour suppressor genes. The miR-17-92 cluster forms a bistable switch with Myc and the E2F proteins [56].

Chapter 5

Pattern recognition methods for stochastics processes

We as humans are "amazing pattern recognizer" and in fact there are many studies that try to find the link between pattern recognition and human intelligence in order to build artificially intelligent machines [2]. This is not the theme of this thesis but it is only to say that despite in the time domain signals of Autorepressor model and Toggle Switch model our powerful brain can recognize some patterns, in this context of stochasticity and, perhaps, non-periodic signal, in order to verify that they are really present, and maybe achieve some useful quantitive measurements, we need some particular tools to investigate such signals. Let us discuss the ones that we have thought to use in this work of thesis.

5.1 The Autocorrelation Function (ACF)

Summing up what said in the first part, by stochastic processes we mean, in a loose sense, systems which evolve *probabilistically* in time [25]. For *stochastic systems* it is not possible to exactly determine the state of the system at later times given its state at current time. Hence, to describe a stochastic system we use the probability that the system is in a certain state. Nevertheless, such a calculation is often difficult, and, if we are considering a measurable quantity $Y(t)$ which fluctuates with time, we usually focus on finding the moments of the probability distribution, such as the *mean* and the *variance*. These two quantities are commonly measured experimentally [1].

However, the mean and the variance do not tell a great deal about the underlying dynamics of what is happening. What would be of interest is a measure of the influence of a value of Y at time t on the value at time $t+\tau$ [25]. Such a quantity is the *autocorrelation*

function, which was firstly introduced as

$$G(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_0^T dt Y(t) Y(t+\tau) \tag{5.1}$$

This is the time average of a two-time product over an arbitrary large time T, which is then allowed to become infinite. It is interesting to compute the autocorrelation function starting from the *spectrum* of the quantity $Y(t)$ that is defined in two stages:

$$y(\omega) = \int_0^T dt e^{-i\omega t} Y(t) \tag{5.2}$$

then the spectrum is defined by:

$$S(\omega) = \lim_{T\to\infty} \frac{1}{2\pi T} |y(\omega)|^2 \tag{5.3}$$

The autocorrelation function and the spectrum are closely connected. By a little manipulation one finds:

$$S(\omega) = \lim_{T\to\infty} \left[\frac{1}{\pi} \int_0^\infty \cos(\omega\tau) d\tau \frac{1}{T} \int_0^{T-\tau} Y(t) Y(t+\tau) dt \right] \tag{5.4}$$

and taking $T \to \infty$ one finds:

$$S(\omega) = \frac{1}{\pi} \int_0^\infty \cos(\omega\tau) G(\tau) d\tau \tag{5.5}$$

This is a fundamental result that relates the Fourier transform of the autocorrelation function to the spectrum. The result may be put in an other form by considering that:

$$G(-\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-\tau}^{T-\tau} dt Y(t+\tau) Y(t) = G(\tau) \tag{5.6}$$

so we obtain:

$$S(\omega) = \frac{1}{2\pi} \int_{-\infty}^\infty e^{-i\omega\tau} G(\tau) d\tau \tag{5.7}$$

hence:

$$G(\tau) = \int_{-\infty}^\infty e^{i\omega\tau} S(\omega) d\omega \tag{5.8}$$

This result is known as the *Wiener-Khinchin theorem*. Its meaning is profound: it tells us that one may either directly measure the autocorrelation function of a signal, or the spectrum, and convert back and forth which by means of the *Fast Fourier Transform* is relatively straightforward. We will discuss about the Fast Fourier Transform in Chapter

5 as an other means to analyze our fluctuating signal $Y(t)$ and we will understand also from simulation results that the autocorrelation function and the fast fourier transform are two different ways to achieve the same information. Hence, they are deeply related.

For the moment we keep discussing merely about the autocorrelation function taking into account that, besides such definition of autocorrelation as time average of a signal, we may also consider the *ensemble average*, in which we may repeat the measurement many times, and compute averages, denoted by $\langle\rangle$. It can be shown that for many systems the time average is equal to the ensemble average; such systems are termed *ergotic* [25]. If we have such a fluctuating quantity $Y(t)$, then we can consider the average

$$\langle Y(t)Y(t+\tau)\rangle = G(\tau) \tag{5.9}$$

this result being the consequence of our ergodic assumption. If the system is ergodic, we must have a constant $\langle Y(t)\rangle$, since the time average is clearly constant. The process is then *stationary* by which we mean that the averages of functions $Y(t_1), Y(t_2)...Y(t_n)$ are equal to those of $Y(t_1+\tau), Y(t_2+\tau)...Y(t_n+\tau)$. In particular, $\langle Y\rangle$ is independent of time. It is often convenient to substract this constant from Y(t) and to deal with the zero-mean process $\bar{Y}(t) = Y(t) - \langle Y\rangle$ and divide by the variance of the stochastic process. The autocorrelation function $G(\tau)$ of a stationary process depends on the *lag* time $|t_i - t_{i+\tau}|$ alone and is not affected by this subtraction. We can denote the lag time with τ and often there exists a constant t_c such that $G(\tau)$ is zero or negligible for $|t_i - t_{i+\tau}| > t_c$; one then calls t_c the autocorrelation time.

We remind that strictly stationary processes do not exist in nature, let alone in the laboratory, but they may be approximately realized when a process lasts much longer than the phenomena one is interested in. One condition is that it lasts much longer than the autocorrelation time. Processes without a finite t_c never forget that they have been switched on in the past and can therefore not be treated as approximately stationary.

To summarize, given the time series $Y(t_1)...Y(t_n)$ of a stationary stochastic process and its mean $\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y(t_i)$, we can first calculate what is called the *autocovariance* function at lag τ

$$G(t_i, t_{i+\tau}) = \frac{1}{n}\sum_{i=1}^{n-\tau}(Y(t_i) - \bar{Y})(Y(t_{i+\tau}) - \bar{Y}) \tag{5.10}$$

Dividing by the variance of the stochastic process, we obtain the autocorrelation function (ACF) at lag τ and so the correlation between series values that are τ intervals apart. A plot of the ACF against τ is known as a *correlogram*.
Hence, the autocorrelation function defines how data points in a time series are related, on average, to the preceding data points [22]. In other words, it measures the self-similarity of the signal over different delay times.

After all this discussion, we can recognize and summerize some autocorrelation function properties:

1. $|G(\tau)| \leq G(0)$
2. $G(-\tau) = G(\tau)$
3. if $Y(t)$ has a periodic component, then $G(\tau)$ will have a periodic component with the same period
4. if $Y(t)$ is ergotic, zero mean, and has no periodic components, then $\lim_{|\tau|\to\infty} G(\tau) = 0$.

We will find these properties in our $Y(t)$ signal analysis.

5.2 Fast Fourier Transform (FFT)

Our time domain signal, that we can name as $f(t)$, i.e. a function of time, can hide interesting features under the hood and some of these can be extracted through the Fourier Transform. By considering the continous function $f(t)$, the Fourier transform is based on the following formula:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt \tag{5.11}$$

the Fourier transform means considering one function of time $f(t)$ and apply a transform, through an integral function, that is the complex exponential $e^{-i\omega t}$ that extracts periodicity from the signal. It is the so called "inner product" that is a generalization of the scalar product in which you project one vector onto an other and you see how much of this vector is along an other one. In this case, you can calculate how much of this function $f(t)$ "is along this function $e^{-i\omega t}$", so how much of this function $f(t)$ contains the type of periodicity $e^{-i\omega t}$. You can do this for different functions that depend on ω and you will get a function which depends on ω that tells you how much is the weight associated to each frequency, hence the frequency domain will tell which is the relative contribution of each frequency into comprising function. We can take *any* function and decompose it into sum of pure waves with different frequencies. The concept was originated by Carl Friedrich Gauss and by Jean-Baptiste Joseph Fourier in the early 19th century.

Eq. 5.11 considers a continous function. Most of the signals you come across in nature are continuous (electrical signals in your body, human speech, any other sound you hear, the amount of light measured during the day, barometric pressure, etc etc), however, whenever you want to digitize one of these analog signals in order to analyze and visualize it on a computer, it becomes discrete. Indeed our time series of molecules are discretized time series and so they are vectors of values. When both the function and its Fourier transform are replaced with discretized counterparts, it is called the discrete

Fourier transform (DFT). The DFT transforms the original N-dim data, i.e. it projects onto the new N-dim space. Indeed, the FFT $y[k]$ of length N of the length-N sequence $x[n]$ is defined as:

$$y[k] = \sum_{n=0}^{N-1} e^{-2\pi i \frac{k_n}{N}} x[n] \tag{5.12}$$

and the inverse Fourier transform is then defined as follows:

$$x[n] = \sum_{n=0}^{N-1} e^{2\pi i \frac{k_n}{N}} y[k] \tag{5.13}$$

In this case, we really are multiplying a signal with a series of sine-waves with different frequencies and so we are able to determine which frequencies are present in a signal. If the dot-product between our signal and a sine wave of a certain frequency results in a large amplitude this means that there is a lot of overlap between the two signals, and our signal contains this specific frequency. This is of course because the dot product is a measure of how much two vectors / signals overlap.

The DFT has become a mainstay of numerical computing in part because of a very fast algorithm for computing it, called the Fast Fourier Transform (FFT). It makes use of the symmetry of the sine and cosine functions and other math shortcuts to get the same result much more quickly.

Thus, the Fourier analysis tells us how much of data is organized as function of *frequency* and no more as function of *time* so it is used to transform a signal from a time domain to a frequency domain.

Fig. 5.1 is an example of a signal that contains four different frequencies at four different times.

Fig. 5.2 is the corresponding frequency spectrum.

The four frequencies, already clearly detectable by eye in the time domain, are all present in the frequency domain. The peaks in the frequency spectrum indicate the most occurring frequencies in the signal. The larger and sharper a peak is, the more prevalent a frequency is in a signal.

However, as in the case of the autocorrelation function, if the signal is not periodic, the Fourier spectrum would not be really informative. The sine function is the same everywhere, no matter where you look. It stretches to infinity in a regular periodic fashion which meakes it inconvient for time series analysis. Indeed we can say that the autocorrelation function and the Fourier transform are two ways to obtain the same information from the signal: the extraction of periodicity from the signal. Computing the autocorrelation function we obtain a graph with the autocorrelation values as function of time; with the Fourier transform we obtain the FFT amplitude as function of frequencies. Converting the values of the autocorrelation peaks from the time-domain to the frequency domain should result in the same peaks as the ones calculated by the FFT. The frequency

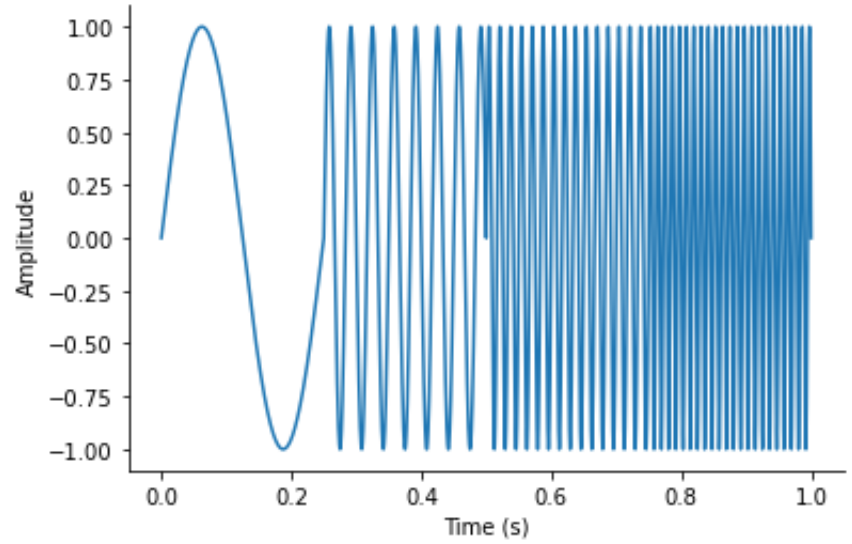

Figure 5.1: Example of time domain oscillating signal which contains four different frequencies at four different times (in particular the frequencies respectively are 4 Hz, 30 Hz, 60 Hz and 90 Hz).

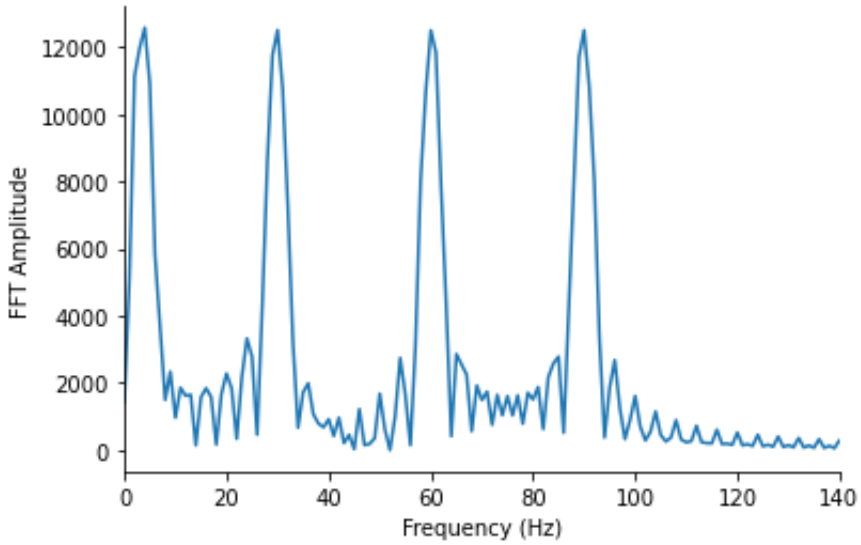

Figure 5.2: Example of FFT applied to the signal in figure 5.1.

of a signal thus can be found with the auto-correlation as well as with the FFT and this was already mentioned in a more "formal" way in section 5.1 as a result of the *Wiener-Khinchin theorem*.

Furthermore, from these figures and from the FFT theory explained before, we can

understand that this approach has an important drawback: the Fourier Transform has a high resolution in the frequency-domain but zero resolution in the time-domain. This means that it can tell us exactly which frequencies are present in a signal, but not at which location in time these frequencies have occurred. It is fundamentally impossible to have both: time and frequency resolution simultaneously. There is always a trade off of information between the two. This is a manifestation of the *Heisenberg uncertainty principle*. Time and frequency are two extremes of these uncertainty: you can either know exactly what a value of a function is at every time point but at the cost of being completely ignorant about what frequencies are fold in. On the other hand, in the frequency domain we know exactly what frequencies are present in that signal but we have no idea about the temporal dynamics of them. A compromise that would sacrify a little bit of frequency resolution and a little bit of time resolution to know something about both is the Wavelet transform.

5.3 Continous Wavelet Transform (CWT)

The Fourier transform uses sine and cosine waves with different frequencies but with the same shape along all the time domain and they repeat periodically back and forth in time. The Wavelet transform instead uses a series of functions called wavelets which have a beginning and an end. The french word wavelet means "a small wave", and this is exactly what a wavelet is. Hence while the sine and cosine waves are infinitely long and stretches to infinity in a regular periodic fashion, the Wavelets are *localized* in time and with a different shape. This allows the wavelet transform to obtain time information in addition to frequency information.

Indeed, since they are localized in time, this transform implies that we can multiply our signal with the wavelet at different locations in time. The wavelet is thus translated from the beginning to the end of the signal (i.e. the time series). This operation is named as *convolution*. In this way, you are localizing along the time series the value of this finite domain function. The process is illustrated in the fig. 5.3:

Figure 5.3: Sketch of Wavelet transform applied to the signal through convolution. Image from [27].

The wavelet can then also be scaled by dilations and contractions. Furthermore, there are different families of wavelets each one characterized by a specific shape. For instance,

we can consider the so called Morlet wavelets that are frequently used for time-frequency analysis of non-stationary time series data, such as neuroelectrical signals recorded from the brain [29]. Fig. 5.4 shows three different pictures that show the same wavelet function represented using different scales.

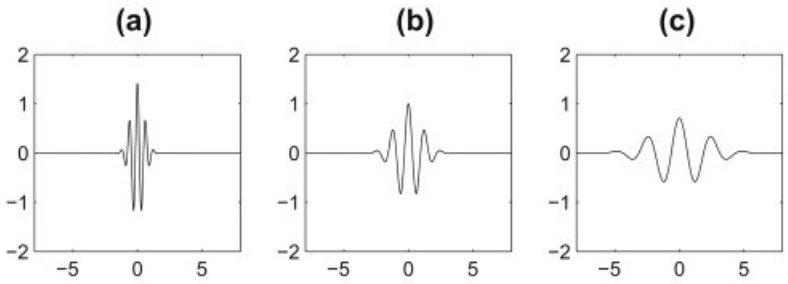

Figure 5.4: Morlet wavelet at three different scales: (a) contracted wavelet, (b) mother wavelet, (c) dilated wavelet. Image from [28].

Using a wavelet function like this you will identify in the time series positive and negative parts.

Mathematically, we can summerize all this with the following formula that represents the Continous Wavelet Transform (CWT):

$$CWT(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} Y(t)\psi\left(\frac{t-b}{a}\right) dt \tag{5.14}$$

In this equation, the parameter a is the scaling factor that stretches or compresses the function. The parameter b is the translation factor that shifts the mother wavelet along the axis. The parameter $Y(t)$ is an integrable signal whose sum is to be multiplied by the translated mother wavelet. And finally, the mother wavelet is denoted by $\psi(t)$, which is a function of the scaling and translation factors just as the result of the continuous wavelet transform is.

Generally speaking, to be considered a proper wavelet, a function $\psi(t)$ has to satisfy two main constraints:

1.
$$\int_{-\infty}^{+\infty} \psi(t)dt = 0 \tag{5.15}$$

2.
$$\int_{-\infty}^{+\infty} |\psi(t)|^2 dt < \infty \tag{5.16}$$

The first one is known as the "admissibility condition" and it tells that the wavelet function should have zero mean. This means that if you take the integral, i.e. the area under the curve, and we sum the positive and negative parts, we should get zero. We can notice that also a sine wave passes the admissibility condition. Secondly, the wavelet function has to have finite energy: if you square the function and compute the area under the curve everywhere from minus infinity to plus infinity it should be a finite number and this is exactly what makes the function localized in time. It means that the wavelet covers a finite area. In fact think about the energy of the sine wave: it is infinite.

As said before, one very known wavelet is the Morlet wavelet that is defined taking the cosine wave of a certain frequency and damping it by multiplying on a Gaussian bell curve:

$$\psi(t) = k_0 \cos(\omega t) e^{-\frac{t^2}{2}} \tag{5.17}$$

this is the *real* component of a Morlet wavelet and has the shape depicted in fig. 5.4. Wavelet itsself is a complex function and we will discuss about this in a few.

Hence, in this case, when doing the Wavelet transform we have to choose the *scale* and the *mother wavelet* to use that will be *translated* and multiplied to the signal. By varying scale and translation parameters we can "scan" our signal with analysing wavelets of different scales to see what frequencies are most prominent around that time point.

The Wavelet transform gives, in any case, information about local periodicities and so how frequencies in a time series data are localized in time. However, the best choice of the parameters just mentioned is able to give more informative results.

We can consider an example of the application of the CWT to the signal rappresented in fig. 5.1 that changes its frequency at four different time steps. A good tool for the CWT representation of our signal is the *scaleogram*. It is a 2D representation of a 1D signal in time domain. The scaleogram, as the name suggests, reports the time on the horizontal axis and the period, i.e. the length of the Wavelet or also called the scale, one the vertical axis. For our convention, the color-level coded scale represents the wavelet coefficients: dark red is associated to high values of wavelet coefficents, while white colored regions represent low values of wavelet coefficients. To clarify, by "wavelet coefficients" we mean the result of the convolution between the time series and the chosen wavelet, just as the FFT amplitude values in the Fourier transform.

On the vertical axis of the scaleogram we put the scale, that is the wavelet length: higher the length of the wavalet is, lower the frequency is; on the contrary, lower the length of the wavelet is, higher the frequency is.

Hence, how do we interpret the scaleogram ? Roughly speaking, the convolution measure the similarity between the wavelet function and the data. If the similarity is high, we will have high values of wavelet coefficients (i.e. red colors); if it is low, it will correspond to low wavelet coefficients (i.e. like white colors). But since wavelets vary from one scale to an other, at certain scales it may or may not have the highest similarity.

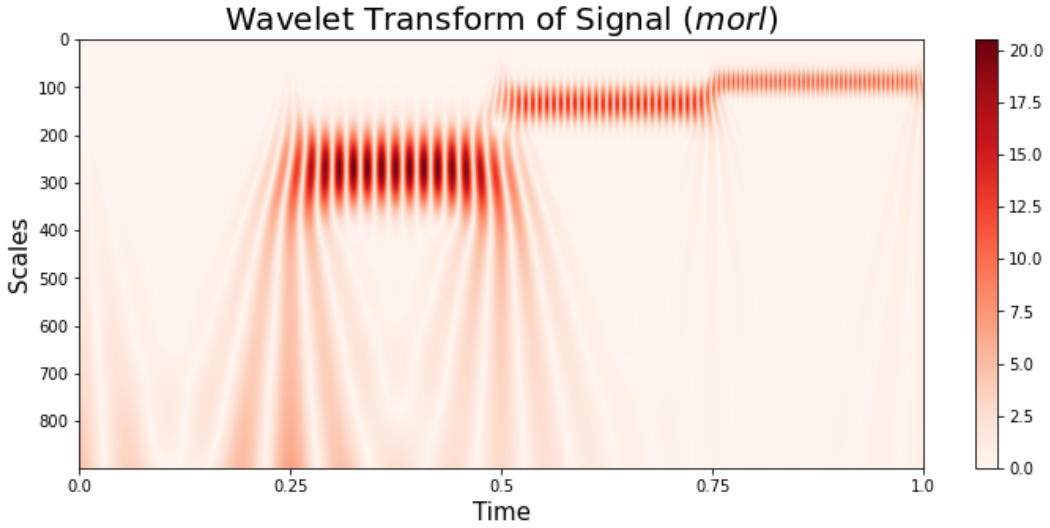

Figure 5.5: Example of Continous Wavelet Transform, using the real part of Morlet wavelet, applied to the signal in figure 5.1.

If at low scales (low values on the vertical axis of the scaleogram) there are high values of the coefficients, then the data have high frequencies; on the other hand, if at high scales there are high values of coefficients, then your data have low frequencies. Thus we can locate this low and high frequency portions by analysing these variations in the values of the coefficients. When there is no match with the signal, the dot product, i.e. the result of integral in Eq. 5.14 is zero; but when the signal approaches the wavelet intrinsic frequency, the function begins like to "resonate" and you get a significant overall contribution when they are in phase, and a significant negative contribution when they are out of phase. When there is match of frequency there is an alternation between positive and negative values. We can realize that Eq. 5.14 is exactly as the dot product between vectors, indeed when vectors have the same direction, the dot product has high value, contrary to when the angle between them is 90° in which case it is zero.

Fig. 5.5 shows the Morlet CWT applied to signal in fig. 5.1 that changes frequency at four time points. Such scaleogram generally shows bands made of an alternation between red and white stripes. In particular there are four bands of this type that are located at lower and lower scale corresponding to the time domain signal that increases its frequency.

However we need to notice that the product in Eq. 5.14 is zero also when the wavelet is exactly in the middle between peaks and troughs of the oscillating signal. It does not mean that there is zero frequency component , but it could be quite the contrary.

This is when complex wavelets come into play. If we consider again the Morlet wavelet defined in Eq.5.17, we obtain the complex Morlet wavelet multiplying it by $e^{i\omega_0 t}$ and so

we obtain:

$$\psi(t) = k_0 \cos(\omega t) e^{i\omega_0 t} e^{-\frac{t^2}{2}} \tag{5.18}$$

In this way the Morlet wavelet is essentially a complex exponent which spins around the circle in the complex plane with a certain constant frequency ω_0 and whose amplitude is modulated by the Gaussian bell curve $e^{-\frac{t^2}{2}}$.

The imaginary part of the Morlet wavelet is like the real part but slightly shifted relative to the cosine. The key idea is to calculate the convolution with both real and imaginary parts. Then our convolution function, for a fixed wavelet scale, will map one real number to the point in the complex plane where the real component is the value of the convolution at that time point with the real part of the wavelet and the imaginary component is the value of convolution with the imaginary part of the wavelet. The power of the frequency, i.e. the intensity of its contribution, at each point in time is given by the distance from the resulting point in the complex plane to the origin which is the *absolute value of the complex number*. In this way we measure the intensity of a particular frequency component as function of time. The resulting function Eq. 5.14 is the complex function, the absolute value of which represents the contribution of a particular frequency around a certain time point. We can again represent it with colors obtaining the scaleogram.

Fig. 5.6 show the scaleograms corresponding to CWT applied to signal in Fig. 5.1 using the complex Morlet wavelet. At first sight, we can understand that this graph is more intuitive than the one obtained through only the real part of Morlet wavelet that was shown in Fig. 5.5.

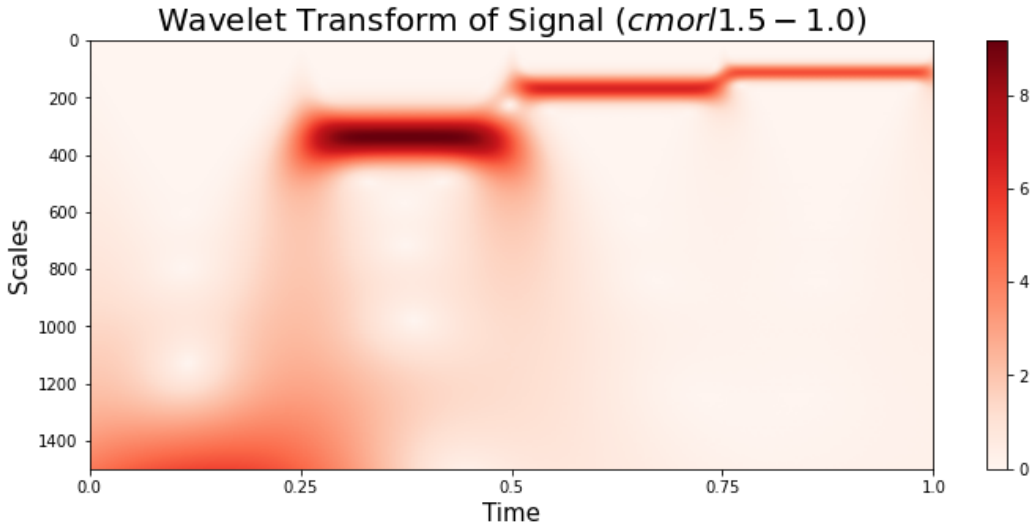

Figure 5.6: Example of Continous Wavelet Transform, using Complex Morlet wavelet, applied to the signal in figure 5.1.

Fig. 5.6 shows four different bands located at four different time points and at four different scales. This time, the bands are uniformly dark red colored and this agrees with our expectations since the time domain signal changes its frequency four different times by increasing its frequency and keeping the frequency constant for a time period at each one of the four times.

Before concluding this overview about wavelets, we need to emphisize that the results may change according to the choice of scale and the choice of the wavelet used. The first one is obvious: by considering a lower range of scales in a situation like this, we would not have detected the lowest frequency present at the beginning of the signal. However, this is true if we use a Complex Morlet Wavelet.

Indeed, in a situation like this, if we apply the Complex Gaussian Wavelet transform to the usual signal in Fig.5.1, we are able to detect low frequency signal by using a lower range of scales.

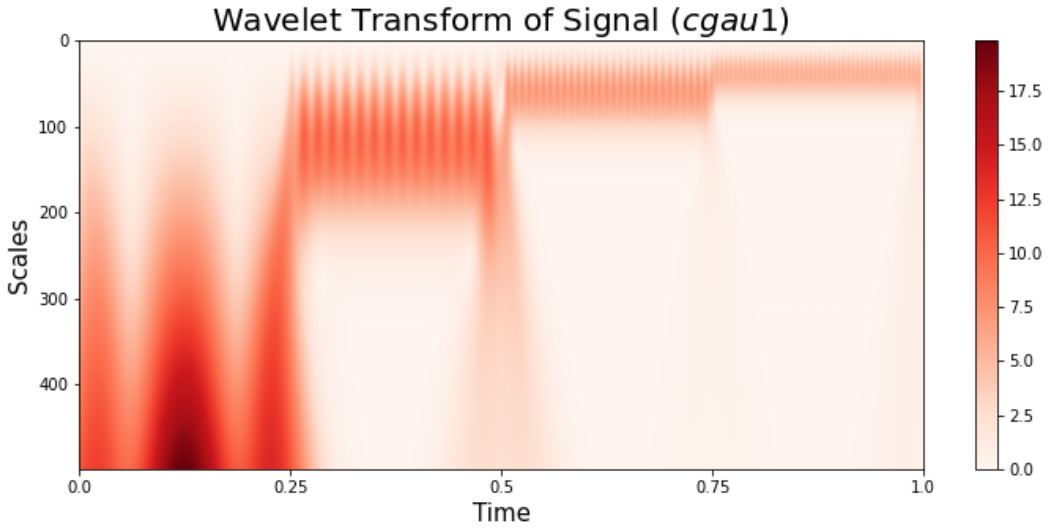

Figure 5.7: Example of Continous Wavelet Transform, using Complex Gaussian wavelet of first order, applied to the signal in figure 5.1.

From these pictures, we can recognize the power of the CWT for dynamically changing signals in recognizing frequencies present in the signal together with their temporal localization. Furthemore, as pointed out previously, we can notice also the importance of the parameters chosen which in this last example is the importance of the choice of the wavelet.

In our genetic circuits analysis we will use an other kind of representation of the continous wavelet transform that is analogous to this but it considers the frequencies of the signal on the y axis instead of the scale of the wavelet. Thus, the corresponding continous wavelet transform to Fig.5.1 is the one is Fig. 5.8.

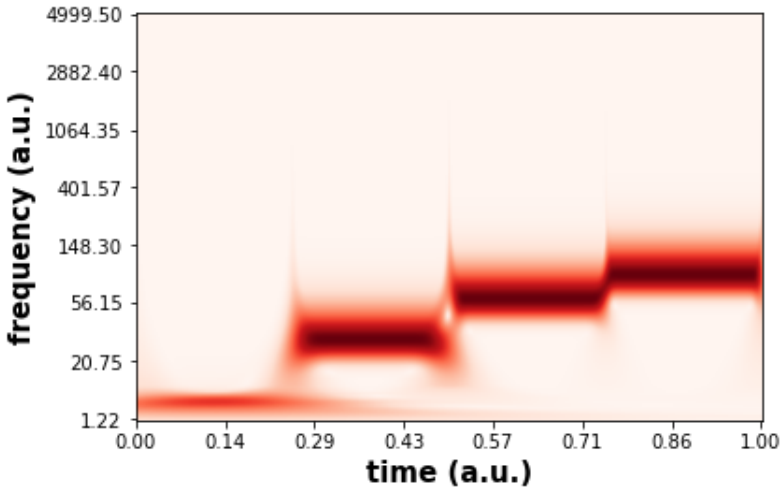

Figure 5.8: Example of Continous Wavelet Transform, using "Generalized Morlet Wavelet" transform, applied to the signal in figure 5.1. Frequencies are reported on the vertical axis while the horizontal axis represents the time.

As we can notice, each red band is located to the precise frequency of the signal, that are 4 Hz, 30 Hz, 60 Hz and 90 Hz.

This representation is even more intuitive. The mechanism of conversion from scale to frequency is not studied in this thesis. We can leave some references to the python module that is used and the relative documentation [66],[67],[68],[69].

5.4 Application of ACF, FFT and CWT to genetic circuits

Up to now, we have studied the basic principals behind the autocorrelation function, the fast fourier transform and the continous wavelet transform but applied to "dummy signals" in order to understand how they work and if they can help us also in the study of our signals that are instead "real" namely they are hidden inside noise. Let us analyse what we can achieve exploiting these tools.

5.4.1 ACF

We can start from studying the first simple protein synthesis model by playing with the effects of regulation as we have done when building the hybrid algorithm. Do we discover a particular autocorrelation shape in the case of "housekeeping" genes products and one for the regulated ?

In the time domain, at first sight, the number of molecules produced and proteins show some particular patterns: the constitutive expression model case has a steady state period (see Fig.4.2) whose signal varies with "smaller amplitudes" with respect to the regulated gene case when the time of residency of gene in active state is higher than the one in inactive state (see Fig. 4.3). Furthermore, as already observed in Chapter 2, the case in which the gene spends more time to be inactive than active (see Fig. 4.4) shows a pattern that remembers the time course analysis for the autorepressor case (see Fig.4.5).

Fig.5.9, Fig. 5.10 and Fig. 5.11 shows the autocorrelation plots for three cases taken into account.

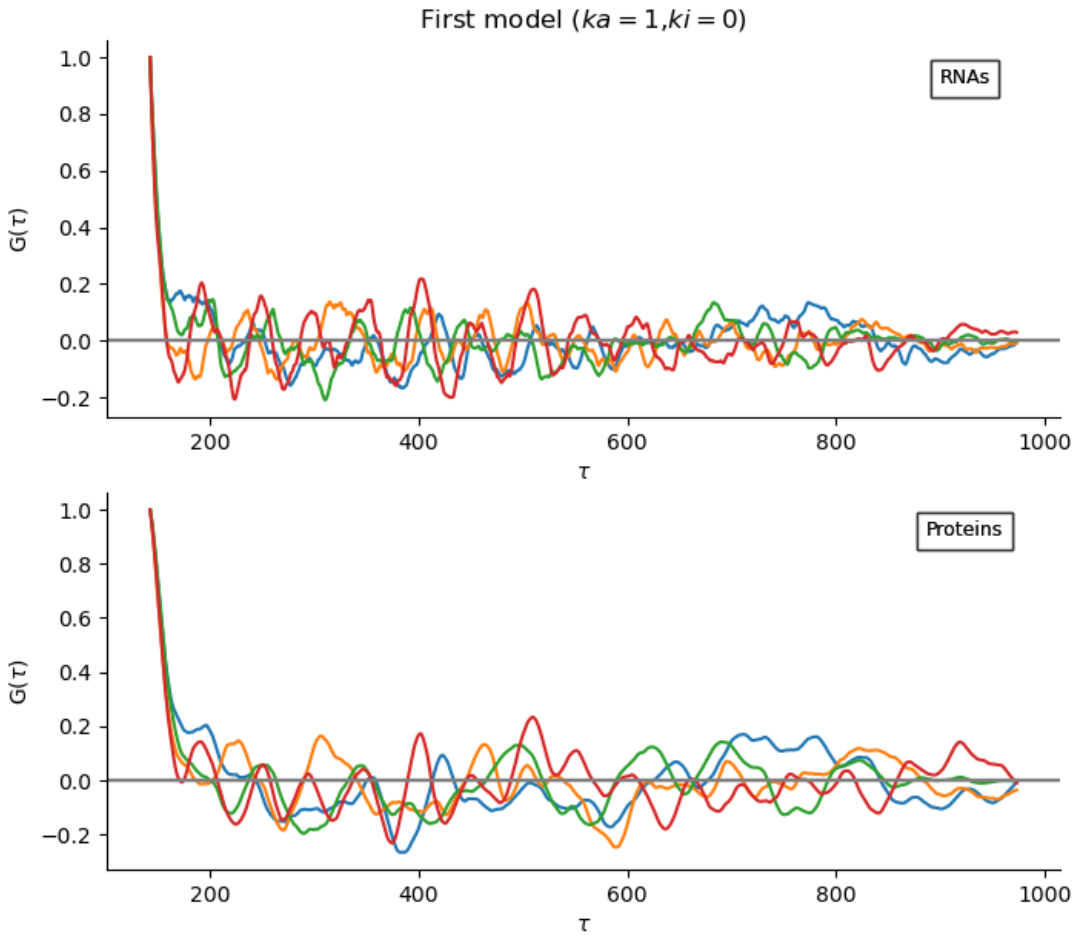

Figure 5.9: Trend of 4 RNAs autocorrelation plot (top) and proteins autocorrelation (bottom) as function of sampling time. The model parameters and time limit are equal to those considered in Fig. 4.2. The number of lags used to calculate the autocorrelation is 10000 with a sampling time dt=0.1.

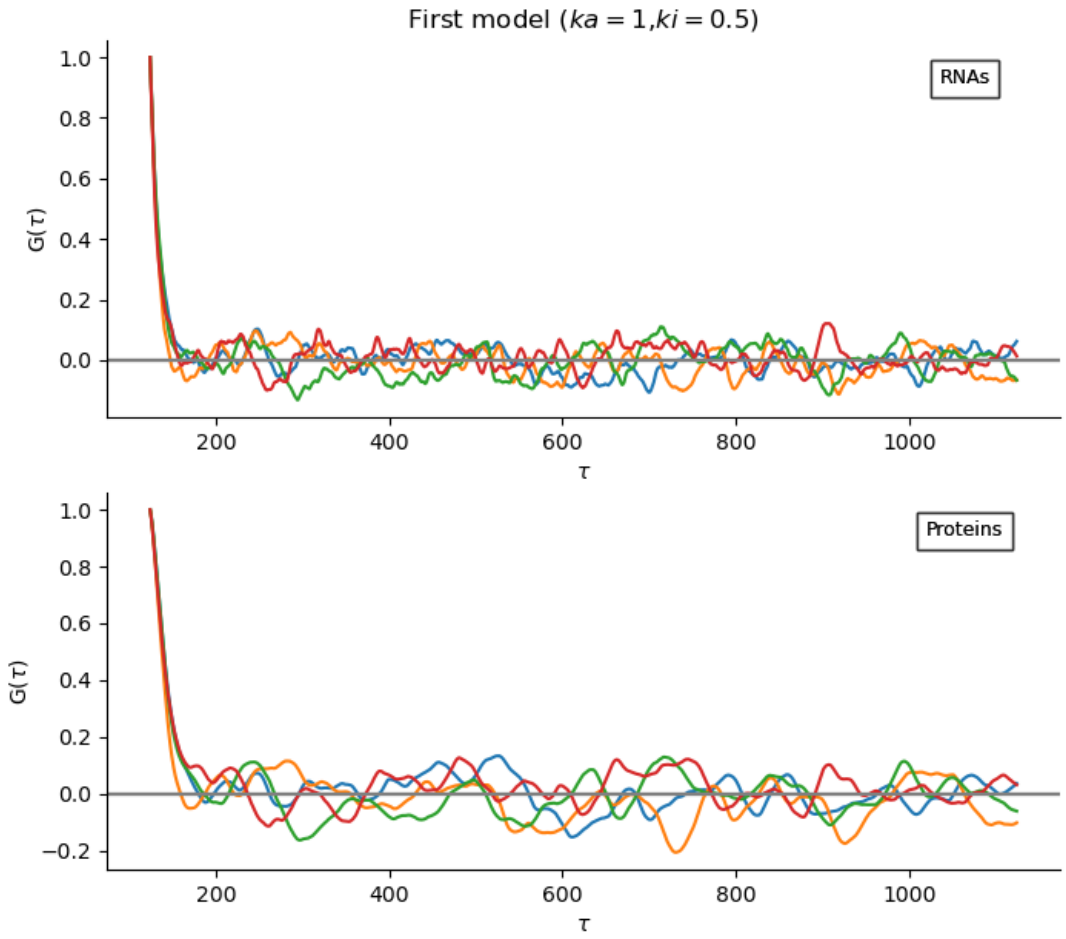

Figure 5.10: Trend of 4 RNAs autocorrelation plot (top) and proteins autocorrelation (bottom) as function of sampling time. The model parameters and time limit are equal to those considered in Fig. 4.3. The number of lags used to calculate the autocorrelation is 10000 with a sampling time dt=0.1.

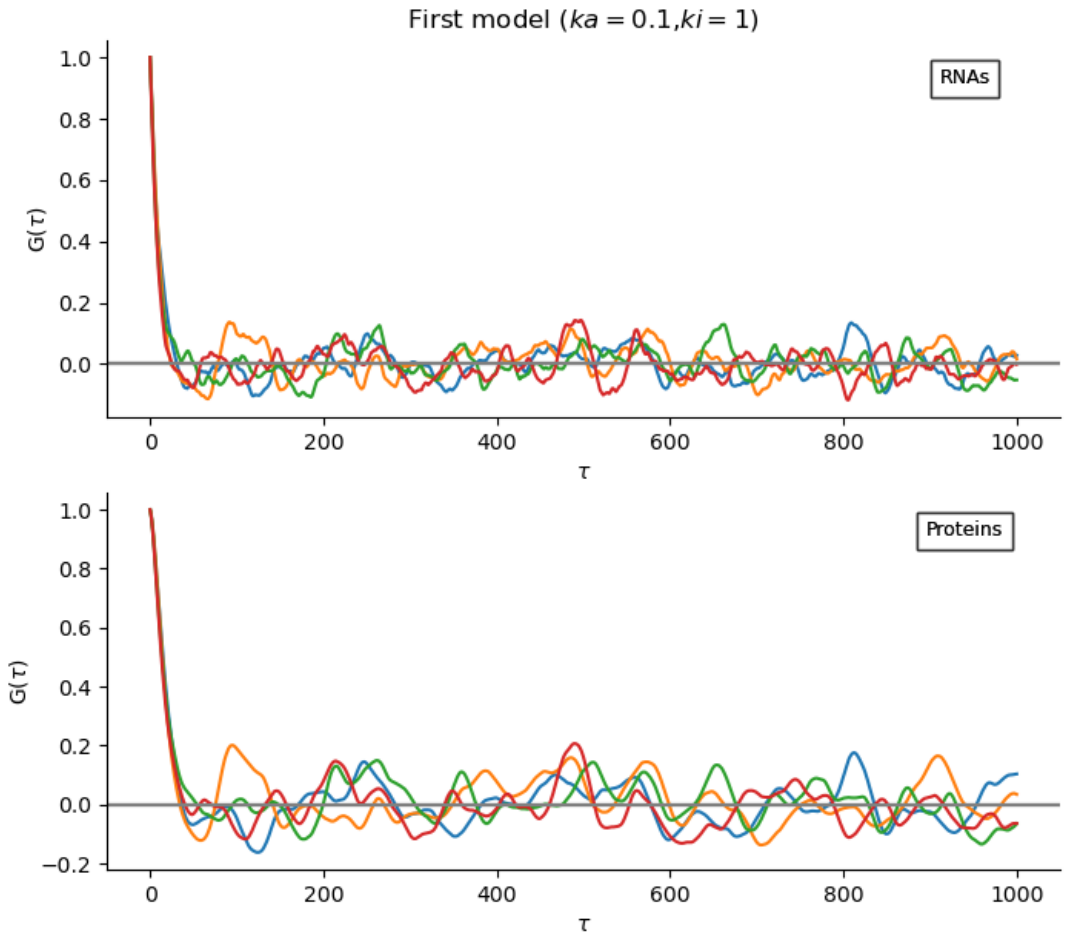

Figure 5.11: Trend of 4 RNAs autocorrelation plot (top) and proteins autocorrelation (bottom) as function of sampling time. The model parameters and time limit are equal to those considered in Fig. 4.4. The number of lags used to calculate the autocorrelation is 10000 with a sampling time dt=0.1.

From Fig.5.9, Fig. 5.10 and Fig. 5.11 we can observe, however, small differences in trends. In all of them we can notice deeper initial negative peaks in RNAs autocorrelation curves with respect to the proteins curves. In case the rate of inactivation of gene is lower than the activation one (Fig.5.11), we can notice that the initial deep is more evident and that it is conserved also for proteins.

However if we consider the autocorrelation plots resulting from the quantiles calculation of 64 simulations shown in Fig.5.12, Fig. 5.13 and Fig.5.14, the three curves are

always the same validating the fact that they come from the same genetic model. What changes is only the type of regulation.

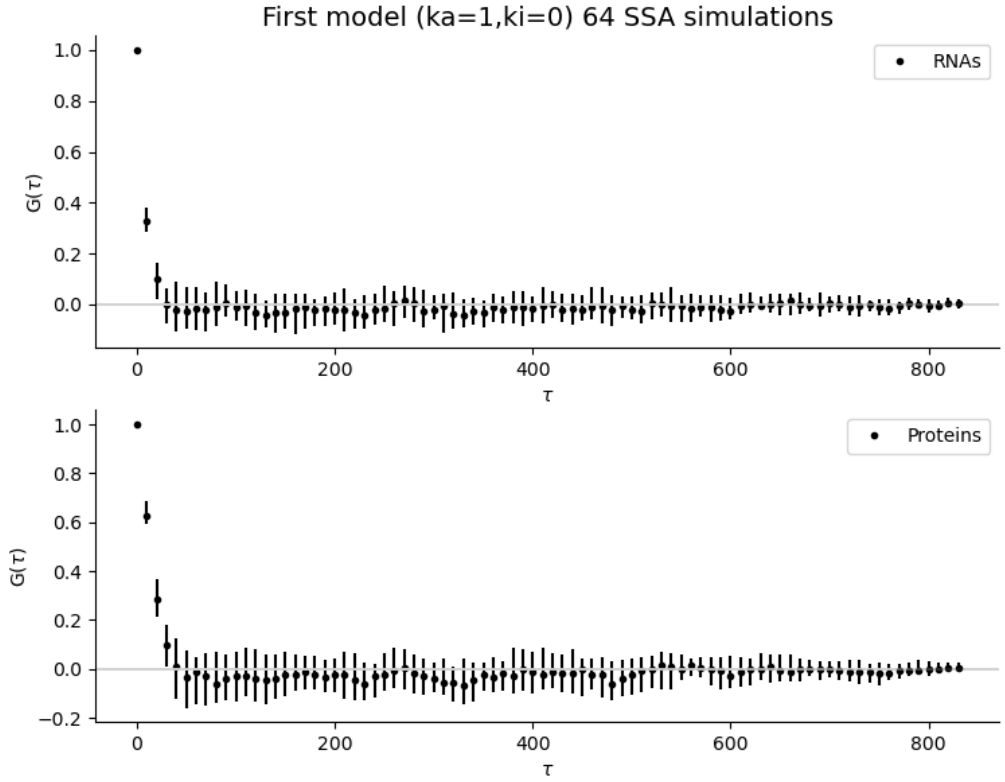

Figure 5.12: Trend of RNAs autocorrelation plot (top) and proteins autocorrelation (bottom) as function of sampling time as a result of 64 simulations. The model parameters and time limit are equal to those considered in Fig. 4.2. The number of lags used to calculate the autocorrelation is 10000 with a sampling time dt=10. The error bars calculated by considering the 25% or 75% quantiles of the several simulations and the points are the median that are associated to each sampling time.

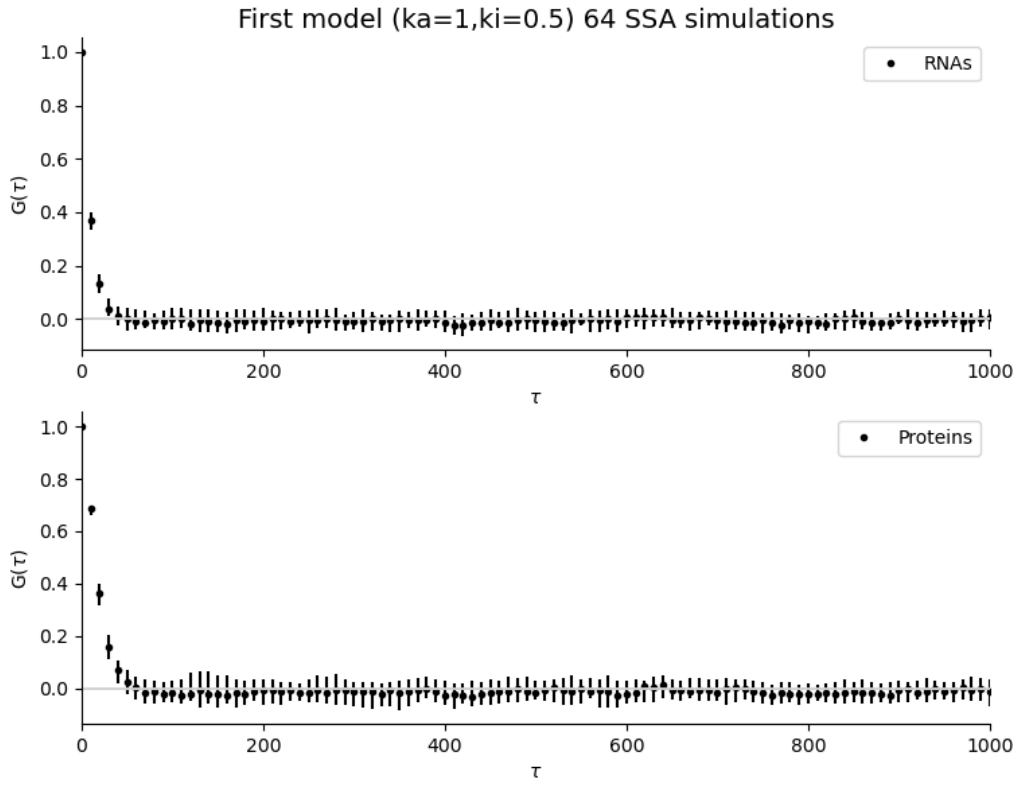

Figure 5.13: Trend of RNAs autocorrelation plot (top) and proteins autocorrelation (bottom) as function of sampling time as a result of 64 simulations. The model parameters and time limit are equal to those considered in Fig. 4.3. The number of lags used to calculate the autocorrelation is 10000 with a sampling time dt=10. The error bars calculated by considering the 25% or 75% quantiles of the several simulations and the points are the median that are associated to each sampling time.

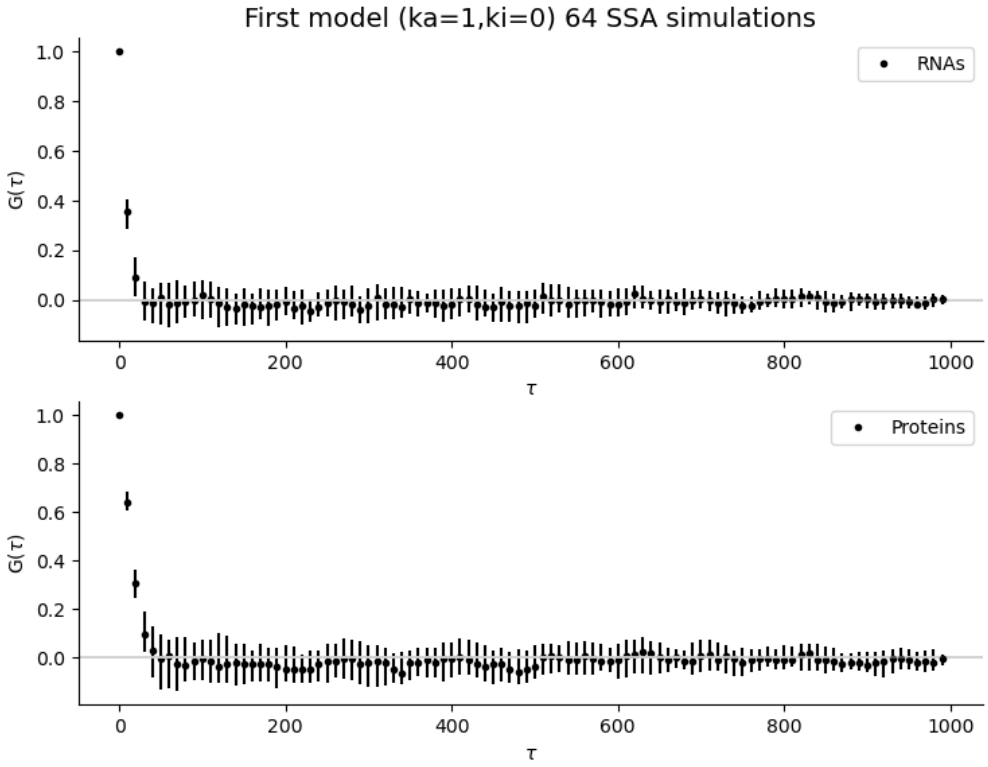

Figure 5.14: Trend of RNAs autocorrelation plot (top) and proteins autocorrelation (bottom) as function of sampling time as a result of 64 simulations. The model parameters and time limit are equal to those considered in Fig. 4.2. The number of lags used to calculate the autocorrelation is 10000 with a sampling time dt=10. The error bars calculated by considering the 25% or 75% quantiles of the several simulations and the points are the median that are associated to each sampling time.

Fig.5.12, Fig. 5.13 and Fig.5.14 show in fact almost the same behavior with the autocorrelation value higher for the first sampling time steps and then it tends to be zero.

We find a different behavior in the second genetic circuit that we have described, that is the autorepressor model.

As we have discussed in Chapter 2, the autorepressor is a little genetic circuit with negative feedback. The gene's proteins product decreases the activation rate of the same gene. We have also observed a typical trend as function of time in Fig.4.5.

At first sight, there is a sawtooth pattern that is repeated over time in the number of RNA molecules vs time trend, whereas the proteins trend is characterised by more "smoothed" edges.

This first impression finds, this time, a sort of confirmation by the autocorrelation plots.

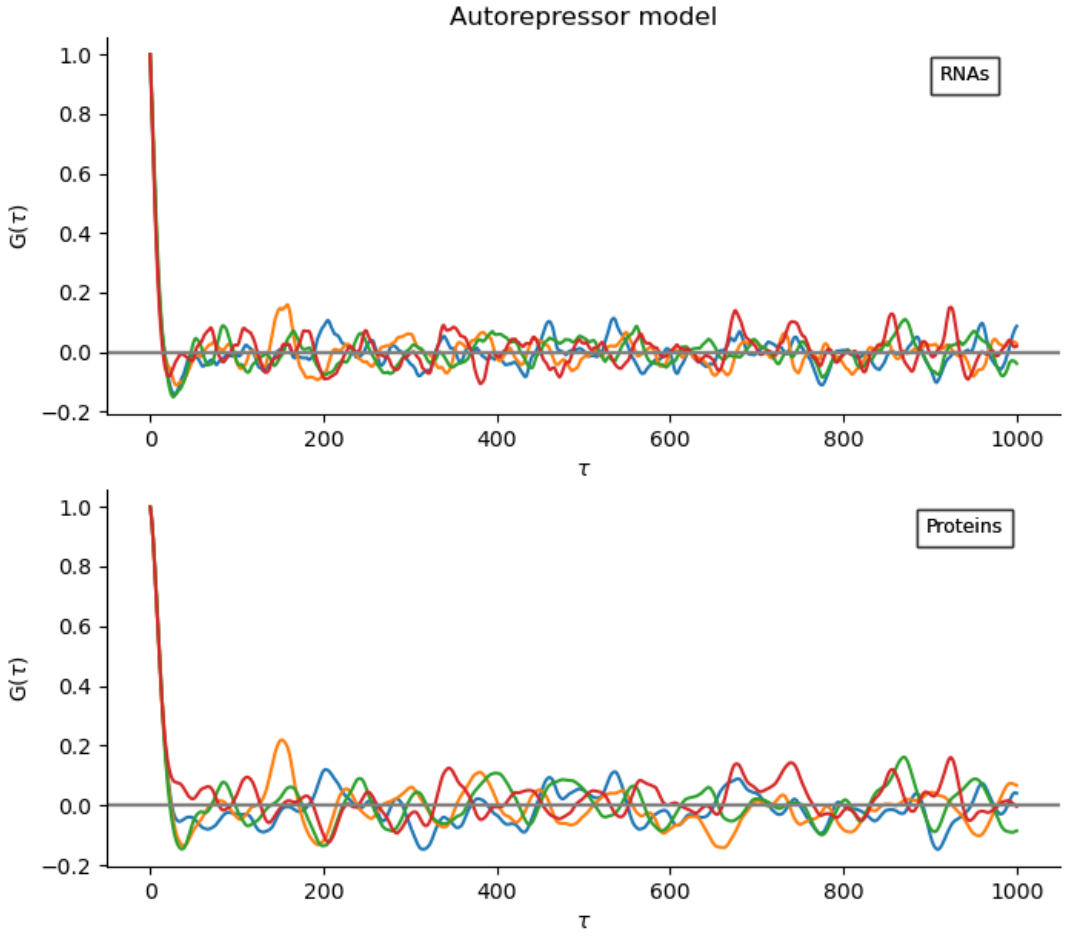

Figure 5.15: Trend of 4 RNAs autocorrelation plot (top) and proteins autocorrelation (bottom) as function of sampling time. The model parameters and time limit are equal to those considered in Fig. 4.5. The number of lags used to calculate the autocorrelation is 10000 with a sampling time dt=0.1.

From Fig. 5.15 we can observe 4 autocorrelation curves calculated for temporal series in Fig. 4.5 changing the random seed. What we can notice is that for the RNAs autocorrelation curves all the curves show a deep at the beginning and then rise and tend to zero values. In case of proteins, not all the 4 autocorrelation curves have this behavior

and this is confirmed when calculating autocorrelation quantiles using 64 simulations (see Fig.5.16).

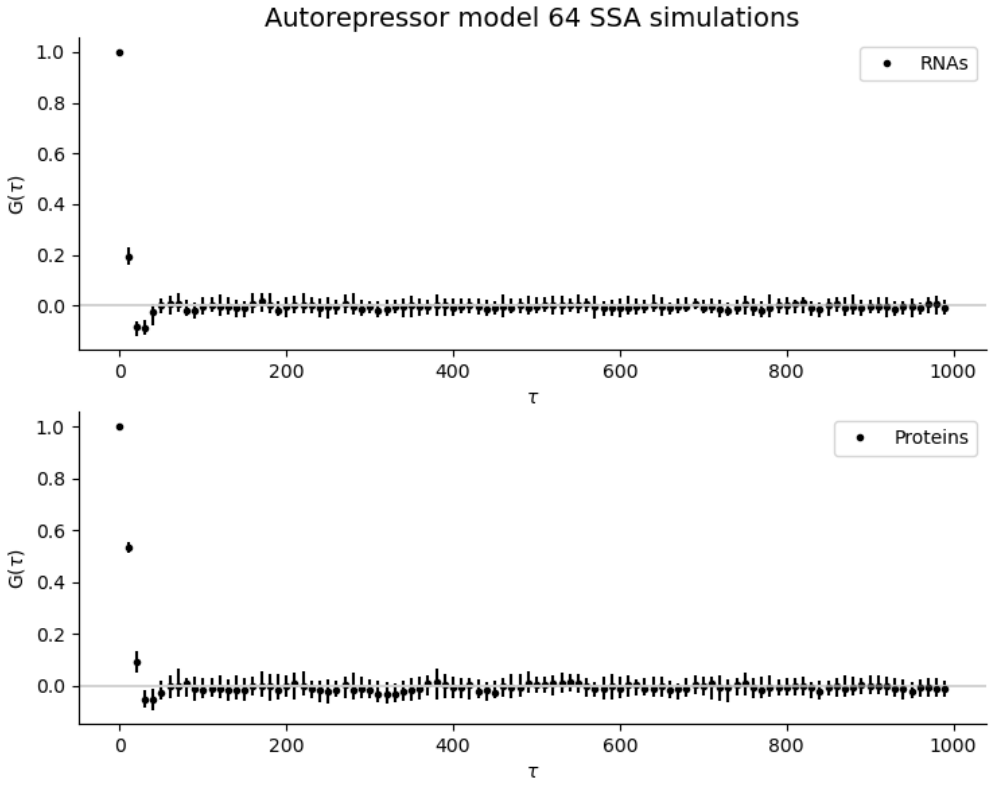

Figure 5.16: Trend of RNAs autocorrelation plot (top) and proteins autocorrelation (bottom) as function of sampling time as a result of 64 simulations. The model parameters and time limit are equal to those considered in Fig. 4.5. The number of lags used to calculate the autocorrelation is 10000 with a sampling time dt=10. The error bars calculated by considering the 25% or 75% quantiles of the several simulations and the points are the median that are associated to each sampling time.

Fig. 5.16 shows RNAs and proteins autocorrelation curves with medians and error bars calculated as result of 64 simulations.

They are both characterized by a deep at the first virtual time points and then it tends to zero. We can clearly see this from the overlapping between the horizontal line at zero and the autocorrelation points or the relative error bars.

We expected the negative deep because of the repressing behavior.

However, in case of proteins autocorrelation function, we find “a less profound deep”

that maybe due to the smoothed time course behavior.

This shows how the autocorrelation function is really sensitive to the shape of the molecular time course behavior.

The last basic genetic circuit analysed is the Toggle Switch model.

In this case we are dealing with expression of two genes and we remind that the proteins products of one gene reduces the rate of activation of an other one.

Let us consider the time autocorrelation function of the two RNAs and proteins products relative to Fig.4.6 for just one simulation shown in Fig.5.17.

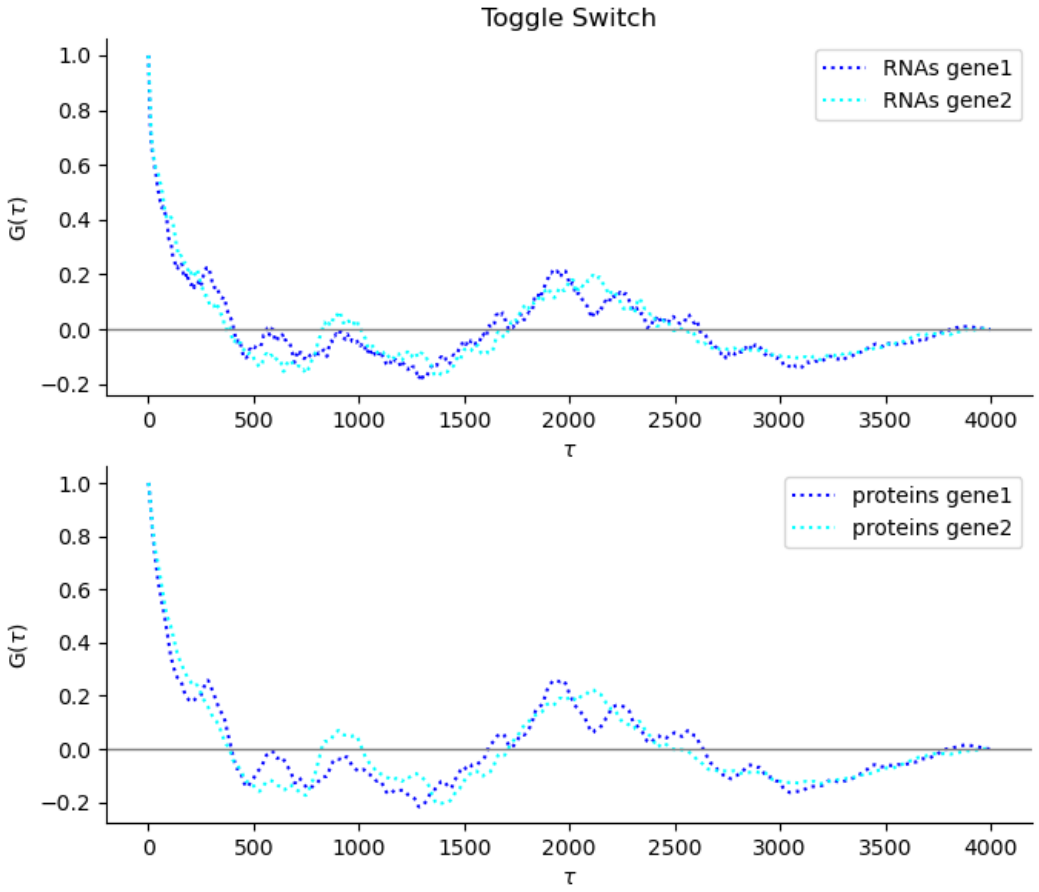

Figure 5.17: Trend of RNAs autocorrelation products of the two genes plot (top) and proteins autocorrelations (bottom) as function of sampling time. The model parameters and time limit are equal to those considered in Fig. 4.6. The number of lags used to calculate the autocorrelation is 10000 with a sampling time dt=0.1.

Notice that the two genes have the same rate constants parameters, hence we expect that the autocorrelation functions of the RNAs molecules for two genes are equal. This is from a mathematical point of view but we can notice that they are not exactly the same due to noise that, in the simulation world, is given by the random number generation of the event and time spent in each state, as explained when presenting the Gillespie's algorithm in section 3.4.1. This similar trends validate that our analysis is robust.

We can point out also that the autocorrelation function shape is much different from the those we have seen in the first simple model and the autorepressor one. It has indeed a much stronger oscillating behavior that is conserved also in proteins autocorrelation functions.

Let us consider now multiple simulations, as we have done for the previous genetic models. We consider the autocorrelation functions of 4 simulations of gene 1 and protein 1.

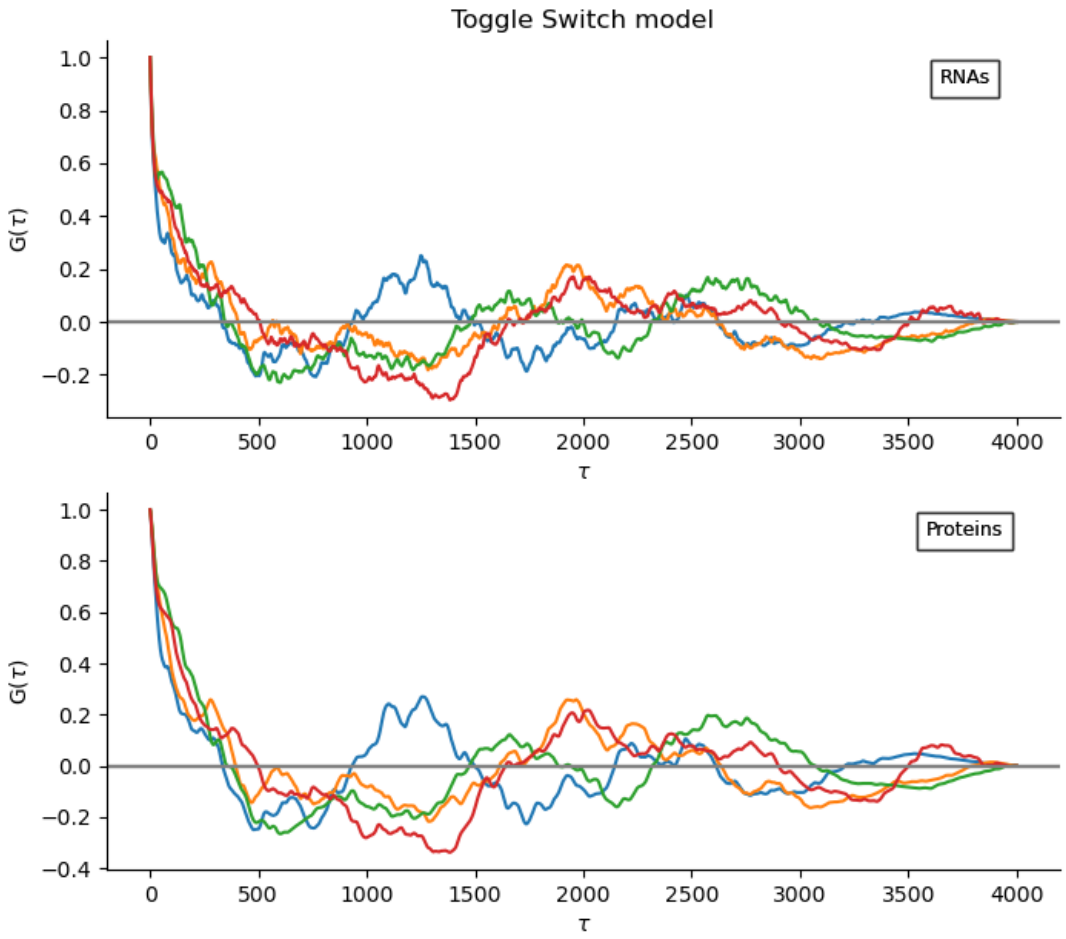

Figure 5.18: Trend of 4 RNAs autocorrelation plot (top) and proteins autocorrelation (bottom) as function of sampling time. The model parameters and time limit are equal to those considered in Fig. 4.6. The number of lags used to calculate the autocorrelation is 100000 with a sampling time dt=0.1.

We can notice that for different random seeds the trajectory has an oscillating behavior but it has not "a common feature" like in the case of the autorepressor where a clear intial deep is shown at the beginning and then for all SSA simulations, all the autocorrelation functions tend to zero. Hence if we consider the usual classical plot of autocorrelation function resulting from the quantiles calculation of 64 simulations, we do not notice any oscillating behavior (see Fig.5.19).

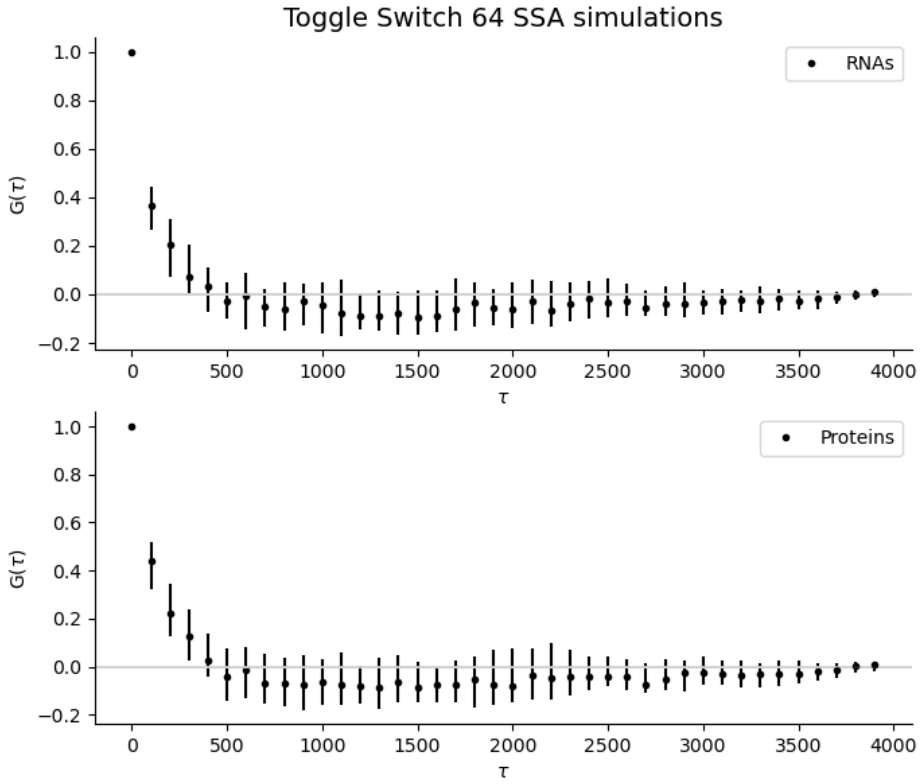

Figure 5.19: Trend of RNAs autocorrelation plot (top) and proteins autocorrelation (bottom) as function of sampling time as a result of 64 simulations. The model parameters and time limit are equal to those considered in Fig. 4.6. The number of lags used to calculate the autocorrelation is 10000 with a sampling time dt=100. The error bars calculated by considering the 25% or 75% quantiles of the several simulations and the points are the median that are associated to each sampling time.

However Fig.5.19 reflects what we see in Fig.5.18 that is without the error bars. Up to the 3000 sampling points the error bars are large then they start to decrease due to the increasing overlapping between simulations.

The shape of such autocorrelation functions in Fig.5.19 would resemble to the first model case, however in the that case, the error bars where indeed much smaller indicating a smaller variation into the autocorrelation trends for each random seed and the median values lay exactly on the horizontal line corresponding to the zero value y-axis.

Hence, from this analysis, we can conclude that the autocorrelation curves are sensitive to the model parameters but also to the type of model. Given these simple genetic

circuits that we have analysed, it is possible to understand by looking at the autocorrelation functions which are the genetic models from which they have been generated.

5.4.2 FFT

The signals that we have seen in Chapter 2, that we have just analysed with the autocorrelation function, are sometimes signals that contain no visually evident periodic components as in case of first model when the gene tends to be more active than inactive (as in Fig.4.2 and in Fig.4.3) while some patterns seem to rise in the other cases, i.e. Fig.4.4, Fig.4.5 and Fig.4.6.

However the frequency spectrum reveals that there is much more to this signal than meets the eye. There is not only random noise for instance in the first two cases that we have mentioned whose frequency spectrum is given by Fig.5.20 and 5.22 for RNAs and Fig.5.21 and 5.23.

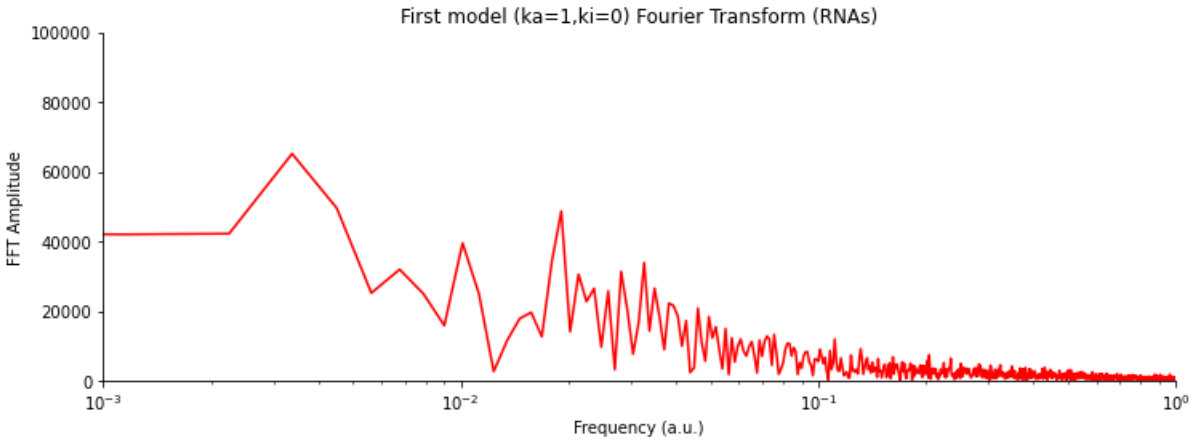

Figure 5.20: Fast Fourier Transform of the number of RNA molecules as function of time in Fig.4.2. The x-axis is in log scale while the y-axis is linear scale.

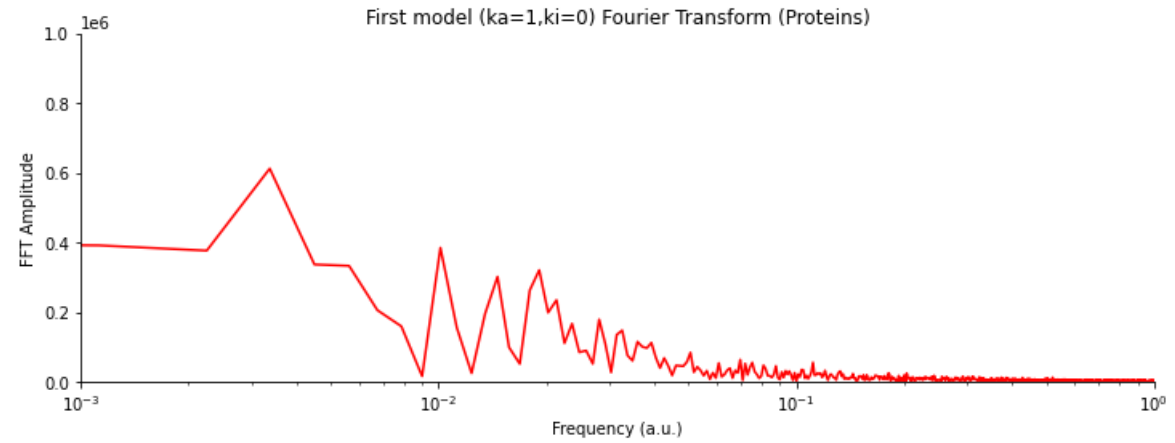

Figure 5.21: Fast Fourier Transform of the number of proteins molecules as function of time in Fig.4.2. The x-axis is in log scale while the y-axis is linear scale.

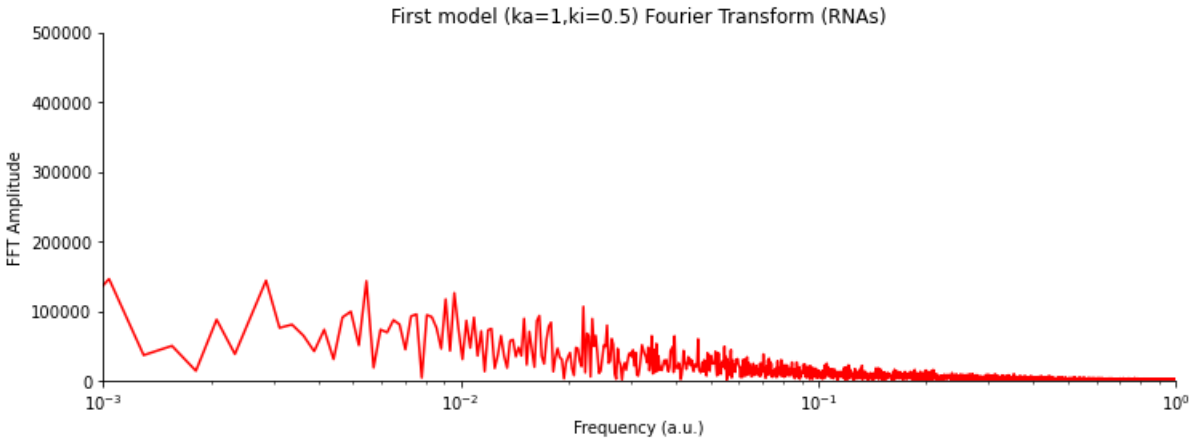

Figure 5.22: Fast Fourier Transform of the number of RNAs molecules as function of time in Fig.4.3. The x-axis is in log scale while the y-axis is linear scale.

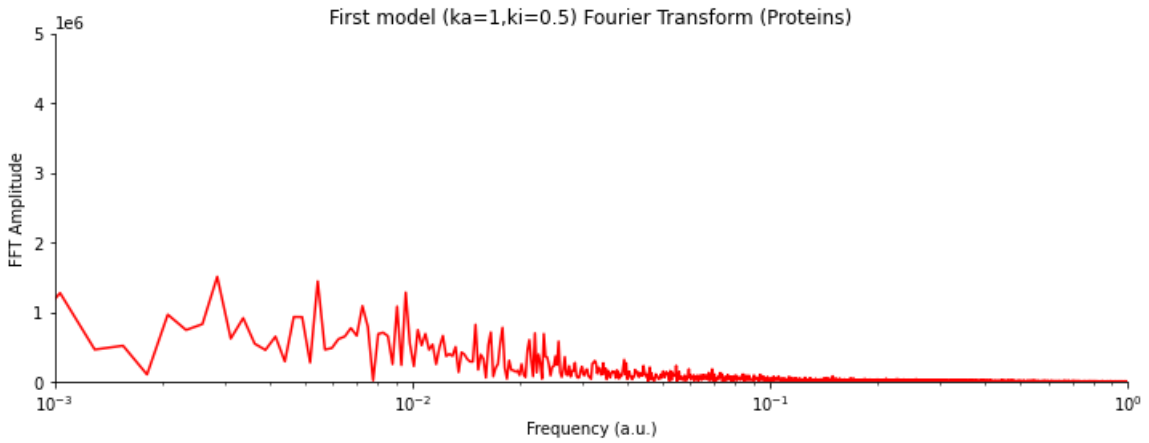

Figure 5.23: Fast Fourier Transform of the number of proteins molecules as function of time in Fig.4.3. The x-axis is in log scale while the y-axis is linear scale.

At low frequencies, Fig.5.20,5.21, 5.22 and 5.23 show sharp peaks indicating a precise periodicity.

Neverthless, this kind of frequency spectrum is common to all the other cases, namely sharp peaks al low frequencies. What changes sometimes is the amplitude of these peaks.

For instance, the amplitudes of the peaks reached in case of the first model when the gene tends to be more inactive than active is comparable to those of the autorepressor case both for what regards RNAs and proteins (i.e. Fig. 5.24,5.25, 5.26 and 5.27).

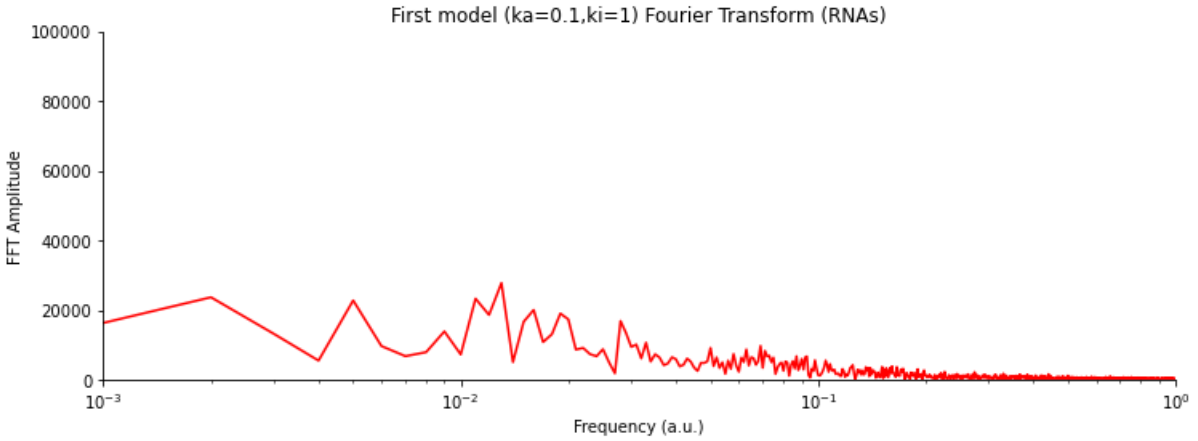

Figure 5.24: Fast Fourier Transform of the number of RNAs molecules as function of time in Fig.4.4. The x-axis is in log scale while the y-axis is linear scale.

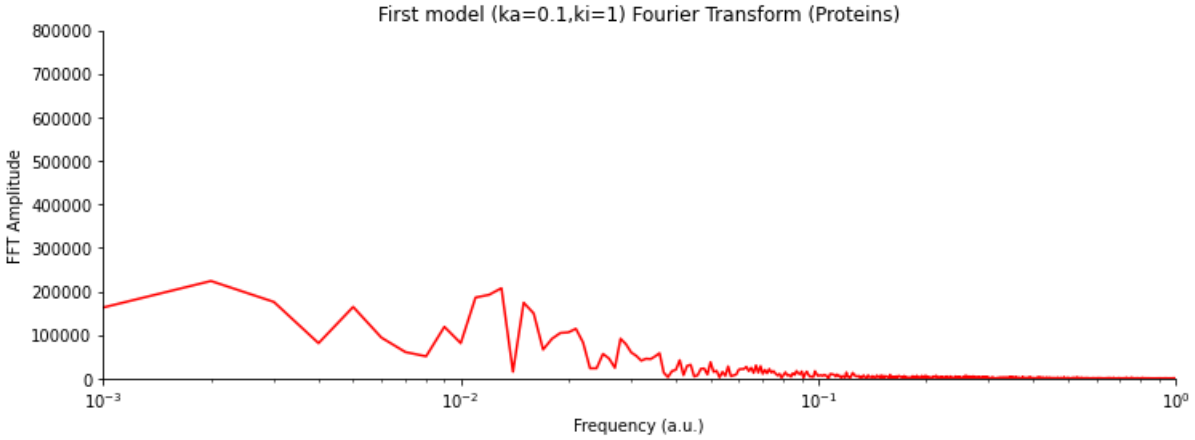

Figure 5.25: Fast Fourier Transform of the number of proteins molecules as function of time in Fig.4.4. The x-axis is in log scale while the y-axis is linear scale.

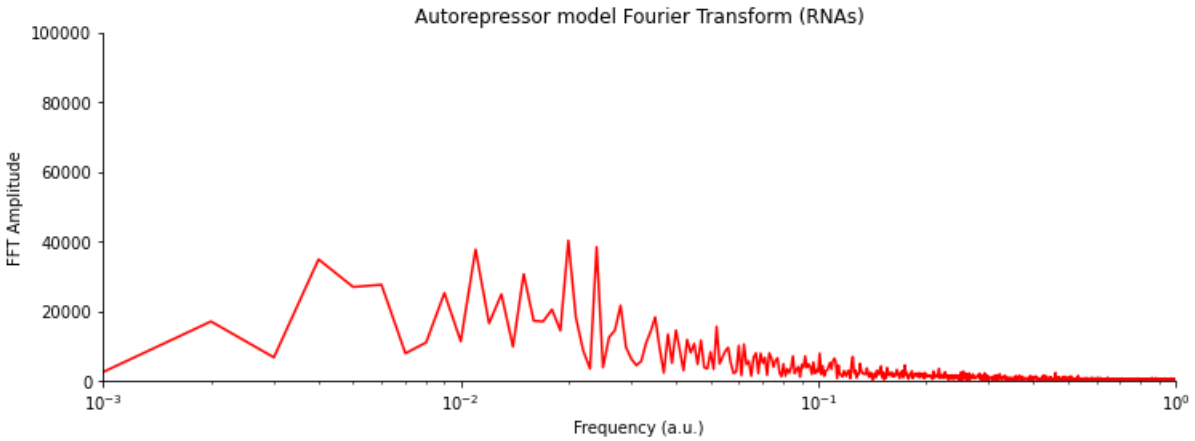

Figure 5.26: Fast Fourier Transform of the number of RNAs molecules as function of time in Fig.4.5. The x-axis is in log scale while the y-axis is linear scale.

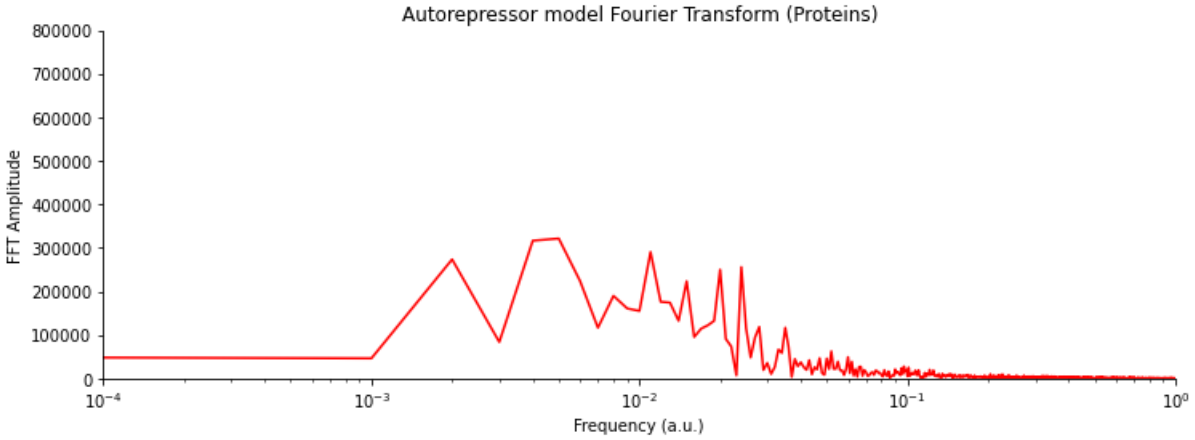

Figure 5.27: Fast Fourier Transform of the number of proteins molecules as function of time in Fig.4.5. The x-axis is in log scale while the y-axis is linear scale.

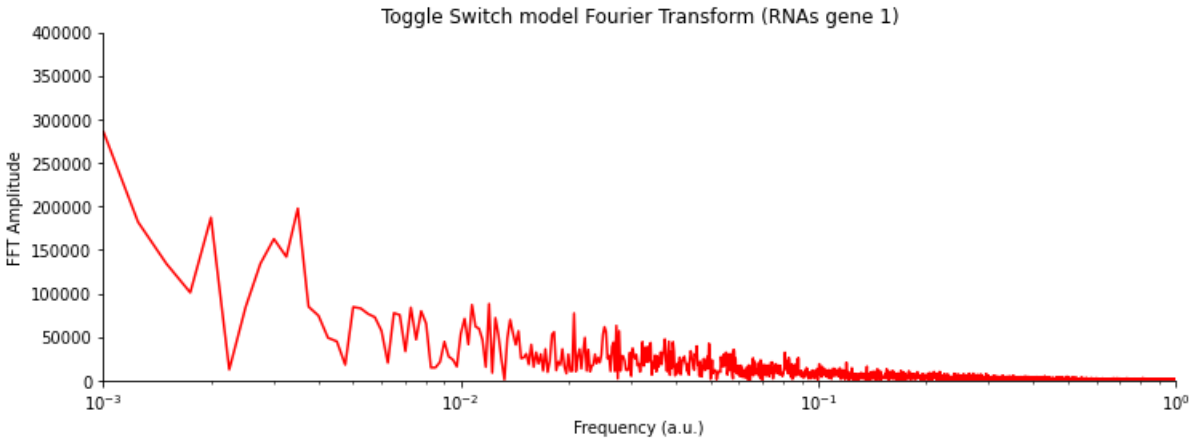

Figure 5.28: Fast Fourier Transform of the number of RNAs molecules as function of time in Fig.4.6. The x-axis is in log scale while the y-axis is linear scale.

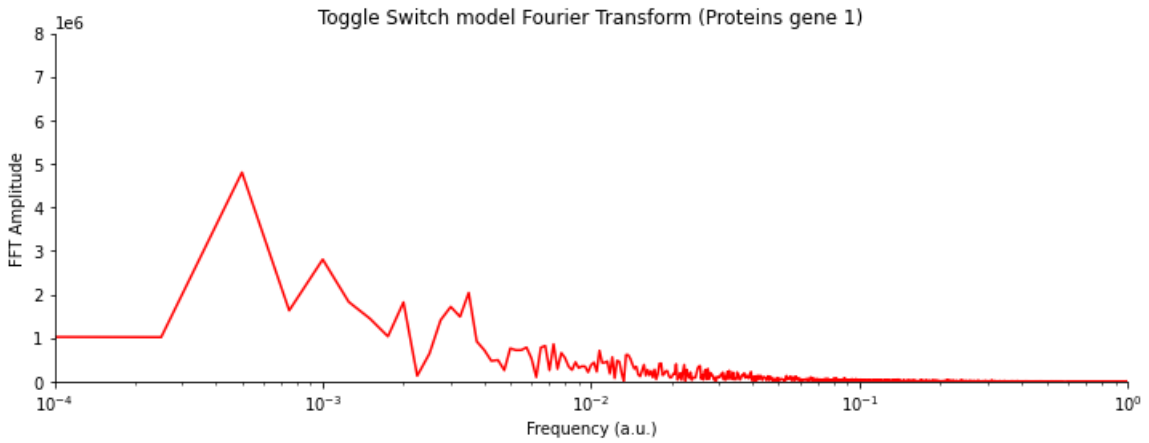

Figure 5.29: Fast Fourier Transform of the number of proteins molecules as function of time in Fig.4.6. The x-axis is in log scale while the y-axis is linear scale.

In general the amplitude of the peaks reached by proteins is higher than those of RNAs frequency spectrum suggesting that those low frequency are more dominant with respect to those of noise.

Moreover, in all cases, after the first sharp peaks, we can see lower peaks that are spread out evenly over the spectrum that is random noise.

The two components are fortunately well separated on the frequency axis, suggesting that low-pass filtering (i.e smoothing) will able to remove the noise without distorting the signal [62].

Note that if the x-axis units of the signal plot had been seconds, the units of the frequency spectrum plot would be Hz; the x-axis of these plots is logarithmic while the y-axis is in linear scale.

Hence, a sort of periodicity arises from these simulated gene expression signals. This is not a surprise since in nature the process of gene expression often shows a precise rythm that maybe caused to environmental factors such as diet, temperature, oxygen levels, humidity, light cycles, and the presence of mutagens can all impact gene expression [63] or to the same gene regulatory code, inscribed in the DNA of each gene, specifies how the production of each gene product will be controlled in space, time and magnitude [64].The gene expression rythm ultimately affects the animal's phenotype. One interesting and much studied synchronization is that between the myocardial contractile response and gene expression of the contractile protein. They show changes with a similar period [65].

5.4.3 CWT

Thanks to the autocorrelation analysis and the fast fourier transform we have achieved important information about our gentic models signals. In both cases we have investigated on the periodicity of the signal and in particular the frequencies present in it through the FFT.

As explained in the theoretical sections about these methods, now we would like to try to get an higher level of information thanks to the Continous Wavelet Transforms. CWT can give information about periodicity and about the frequencies present in the signal as function of time in which the signal is present.

Most of the time, the structure of data is hidden behind the noise so we need this precise mathematical operation with which we can look through the noise and quantify the structure present in the signal. Wavelets are something like could do this blurring of vision and zoom in and out of the signal to pull out the patterns like a kind of mathematical microscope.

As in the ACF and FFT analysis, let us look at the CWT of our genetic models signals starting from the first simple protein synthesis model shown in Fig. 5.30 and 5.31 for the case of constituive expression and Fig. 5.32 and 5.33 in case the gene tends to be more active than inactive.

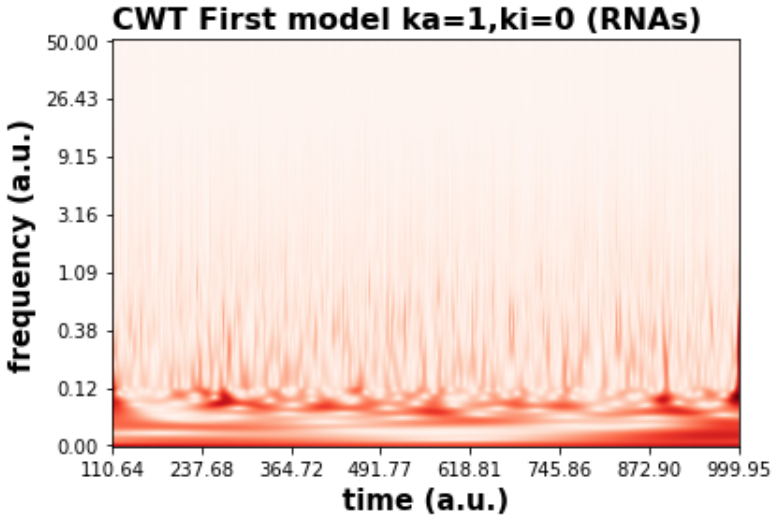

Figure 5.30: Example of Continous Wavelet Transform, using "Generalized Morlet Wavelet" transform, applied to the RNAs molecules signal in figure 4.2. Frequencies are reported on the vertical axis while the horizontal axis represents the time.

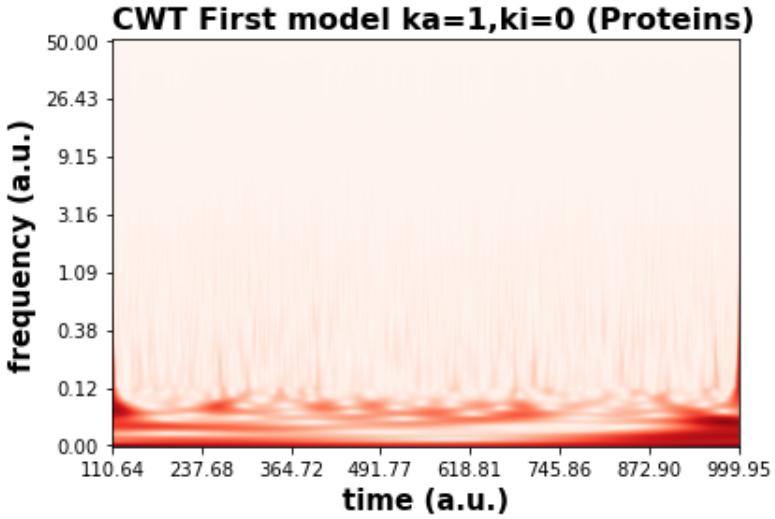

Figure 5.31: Example of Continous Wavelet Transform, using "Generalized Morlet Wavelet" transform, applied to the proteins molecules signal in figure 4.2. Frequencies are reported on the vertical axis while the horizontal axis represents the time.

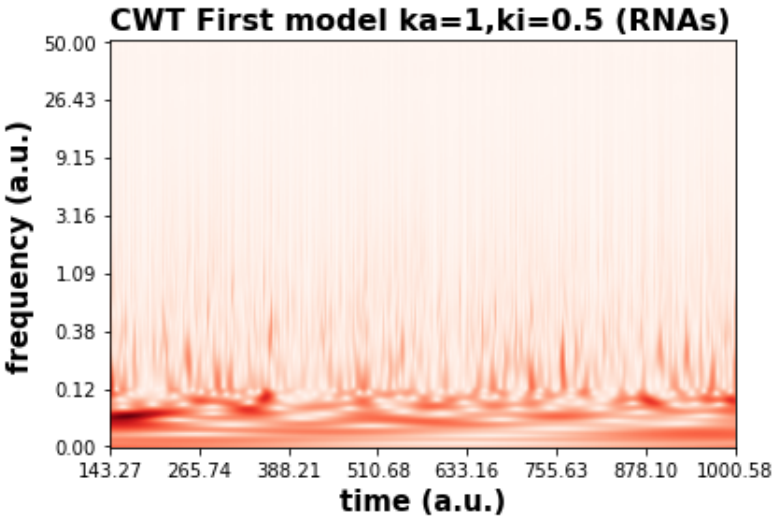

Figure 5.32: Example of Continous Wavelet Transform, using "Generalized Morlet Wavelet" transform, applied to the RNAs molecules signal in figure 4.3. Frequencies are reported on the vertical axis while the horizontal axis represents the time.

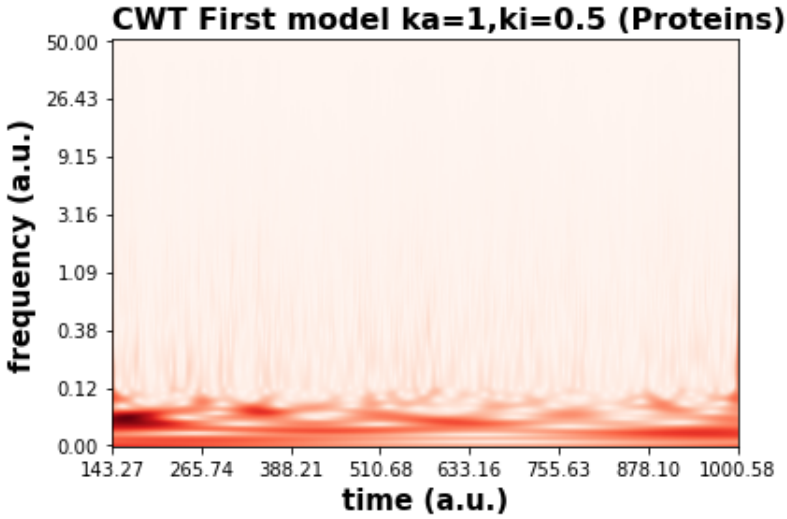

Figure 5.33: Example of Continous Wavelet Transform, using "Generalized Morlet Wavelet" transform, applied to the proteins molecules signal in figure 4.3. Frequencies are reported on the vertical axis while the horizontal axis represents the time.

First, note that the time does not start from zero because of the removed warmup-period.

Then, what we can immediately notice from the CWT is that in the cases in which the gene tends to be active, the CWT shows a more uniform horizontal colored red bar at low frequencies suggesting the presence of a signal behind noise that has a constant frequency. This is not the case of the gene that tends to be more inactive than active or the autorepressor signal (see Fig. 5.34, 5.35, 5.36 and 5.37). In this case we can see red long spots that are located at precise time points. In particular for the autorepressor case they are more regularly located in time. Ideed, in case of autorepressor RNAs signal (Fig.5.36) we could even estimate this regularity in time since red elongated spots are located at approximately every 300 seconds (if we consider the unit of measure of time in seconds). This is very interesting if one wants to study the gene expression rate and its dependences.

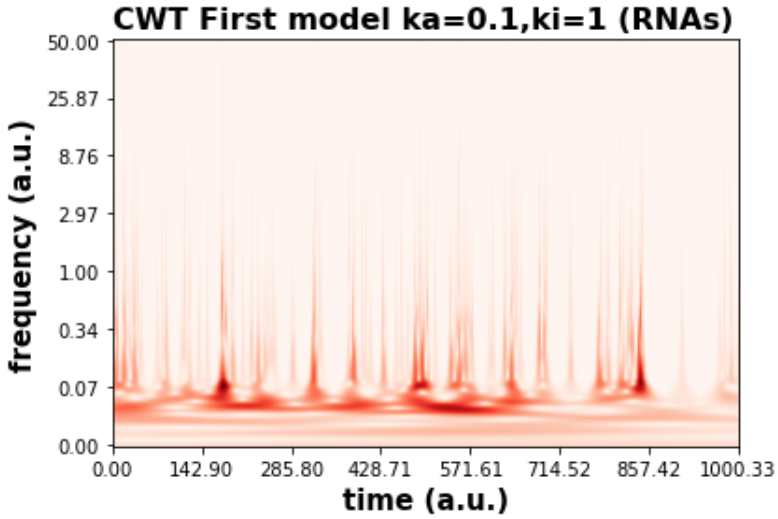

Figure 5.34: Example of Continous Wavelet Transform, using "Generalized Morlet Wavelet" transform, applied to the RNAs molecules signal in figure 4.4. Frequencies are reported on the vertical axis while the horizontal axis represents the time.

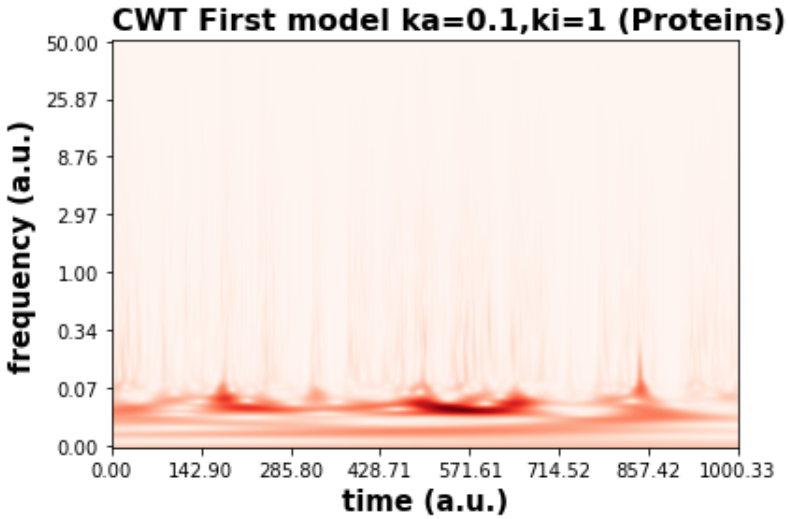

Figure 5.35: Example of Continous Wavelet Transform, using "Generalized Morlet Wavelet" transform, applied to the proteins molecules signal in figure 4.4. Frequencies are reported on the vertical axis while the horizontal axis represents the time.

Anyway, any pattern that we find is located at low frequencies, in the order of 10^{-1} or 10^{-2} and this was expected from the Fast Fourier transform analysis.

We do not have scanned also the toggle switch results by means of CWT because

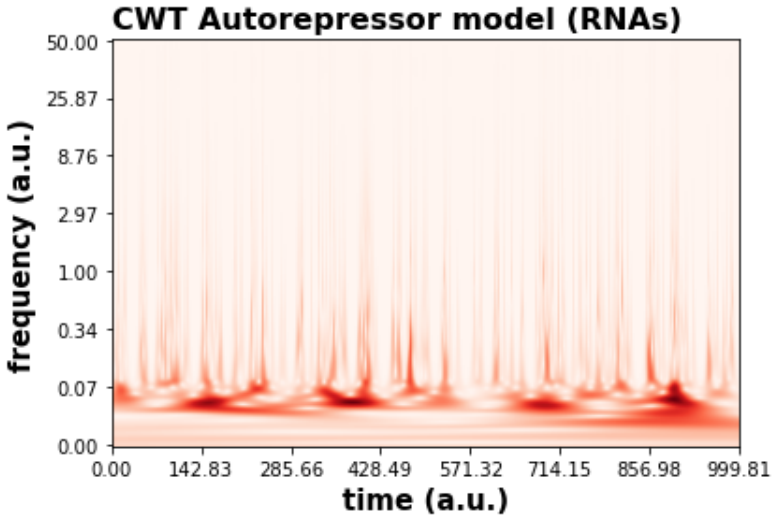

Figure 5.36: Example of Continous Wavelet Transform, using "Generalized Morlet Wavelet" transform, applied to the RNAs molecules signal in figure 4.5. Frequencies are reported on the vertical axis while the horizontal axis represents the time.

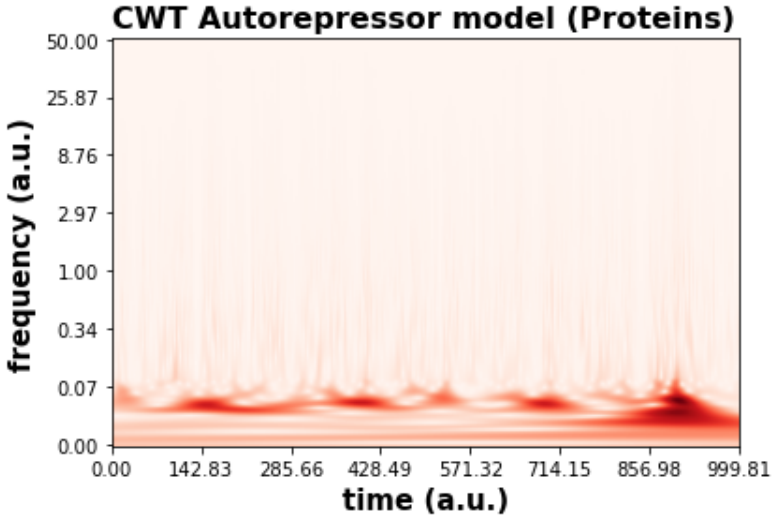

Figure 5.37: Example of Continous Wavelet Transform, using "Generalized Morlet Wavelet" transform, applied to the proteins molecules signal in figure 4.5. Frequencies are reported on the vertical axis while the horizontal axis represents the time.

of lack of computational resources. As explained before, the wavelet transform is computed by a repeated convolution of the signal with the chosen wavelet as the wavelet is translated across the time dimension, in order to probe the time variation, and as

the wavelet is stretched or compressed, in order to probe different frequencies. Because two dimensions are being probed, the result is naturally a 3D surface (time-frequency-amplitude) that can be displayed as we have done, i.e. as a time-frequency contour plot with different colors representing the amplitudes at that time and the frequency. Of course this calculation requires greater execution times. In this thesis work a laptop of CPU 1.10 GHz and RAM 4,00 GB is used and this is not enough for CWT toggle switch analysis. However, in general with modern fast processors and great memory capacity this is unlikely to be a problem.

Part III

Deep Neural Networks for parameters estimation

Chapter 6

Artificial Neural Networks (ANN) as data-driven modeling

The regression problem is how to model one or several dependent variables/responses, Y, by means of a set of predictor variables, X [41].

This problem can be faced in the data-driven modeling framework that is often more suitable to describe the real-world systems that are associated with complexities. A data-driven model is based on the analysis of the data about a specific system. The main concept of data-driven model is to *find relationships between the system input and output variables without explicit knowledge of the physical behavior of the system* that is, in few words, the opposite of what we did in the previous sections when modeling genetic circuits. Unlike first principles models, data-driven models make no attempt to model the internal features of the system. Instead, they focus on matching the input-output behavior to observational data.

The input-output data available from a system can be used to derive different forms of computationally efficient data-driven models for the purpose of prediction, state estimation, monitoring, and other process systems engineering applications. The rapid increase in the availability of data of physical systems has stimulated the development of many data-driven methods for modeling and prediction.

In this thesis we use Artificial Neural Networks (ANN) as a method of data-driven modeling class. We use as input data the autocorrelation function whose shape depends on the system parameters and the output data are, indeed, the constant rates of reactions. Hence, it is a matter of building a *multi-output regression model* using a supervised deep neural network. Then this can be exploited for the purpose of state estimation.

First we will describe the basics artificial neural networks concepts and then we will apply them to the data just mentioned that regard the three genetic models we studied. The results are both important for the building of a data-driven model that is applicable to experimental data and also the building of such deep learning model reveals biological important features of the simple genetic circuits we analysed. Even this last feature can

be considered when dealing with data not coming from simulations.

6.1 Artificial Neural Networks: the basic principles

Artificial Neural Networks are algorithms that mathematically model the neurophysiological structure of human brain.

Human brain is full of neurons and so an artificial neural network is made of many computational units that play the role of neurons. A biological *neuron* receives inputs, transforms it and relays a signal to the other at every step. In biophysical terms, this signal is the action potential that is an electrical signal transmitted by a neural cell across the *axon*.

An artificial neural network is made of a similar system. There are computational units (or even called "nodes") and the *strengths* of the interconnections are represented by *weights*, in which the learned information is stored.

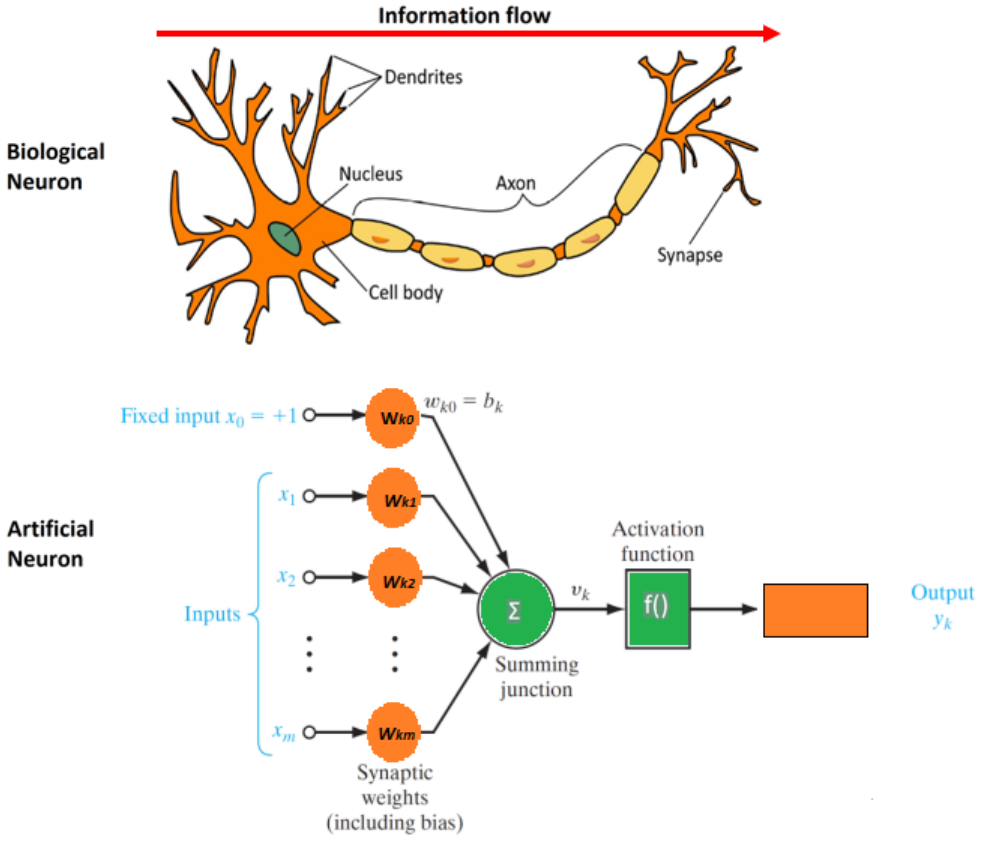

Figure 6.1: Sketch of biological neuron (above) and artificial neuron (below).

Fig. 6.1 shows both the biological and artificial neuron. Hence, the model of a neuron is composed of three basic elements:

- A set of **synapses**, or connecting links, characterized by **weights** or strength (i.e. the efficiency of the propagation of signals from one cell to another [75]), and a signal x_i at the input of synapse i connected to a neuron k. It is multiplied by the synaptic weight w_{ki} (k-*th* neuron, i-*th* input component).
- An **adder** to sum input signals weighted by synaptic strengths (linear integration).
- An **activation function** for limiting the amplitude of the output of a neuron.

Thus, in few words, we can model, from a mathematical point of view, a neuron as a set of inputs which are processed with a set of weights into the computational unit of our neuron. This is the simple Rosenblatt Perceptron model whose output is a linear combination of the inputs x_i weighted by the weights w_{ki}:

$$\nu_k = \sum_{i=1}^{m} w_{ki} x_i \tag{6.1}$$

and then processed by an activation function:

$$y_k = f(\sum_{i=1}^{m} w_{ki} x_i) \tag{6.2}$$

The widely used ANN paradigm is a *multi-layered feed-forward network* (MFFN) with multi-layered perceptron, mostly comprising three sequentially arranged layers of processing units.

The MFFN provides a mapping between an input (x) and output (y) through a nonlinear function (the previous mentioned activation function) f as $y = f(x)$.

The three layered MFFN has input, hidden, and output layers, and each layer comprises of its own nodes. All the nodes in the input layer are connected using weighted links to the hidden layer nodes, and similar links exist between the hidden and output layer nodes. *The weights represent the state of the knowledge* and are adjusted during the learning stage to improve the network performance.

Usually, the input and hidden layers also contain a bias node with its output as unity.

The nodes in the input layer do not perform any numerical processing. All numerical processing is done by the hidden and output layer nodes. Fig.6.2 shows the structure of a typical multinput-multioutput neural network.

The interconnections between the layers of the network and the transfer functions of the neuron processing functions represent the distributed relationships between input and output data. This unique arrangement can acquire some of the neurological processing ability of the biological brain, such as *learning and drawing conclusions from experience.*

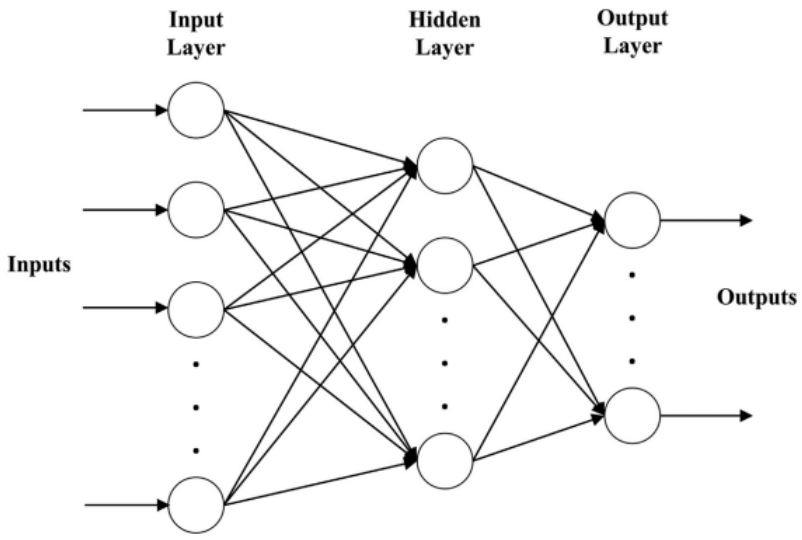

Figure 6.2: Sketch of a typical neural network structure. Image from [41]

6.2 Regressor ANN and its components

There are many categories of neural networks models. They map input patterns to their associated output patterns by learning from examples. The learning paradigm that we will exploit is the supervised learning.

This type of learning is acquired by training the network with a sequence of input and output data and comparing the network predictions with the expected responses.

Network interconnection weights are adjusted according to the learning algorithm and training is continued until the network provides desired responses. NN learning finds a function f that matches the example data.

However, we have to consider that neural network is a "black box" in the sense that while it can approximate any function, studying its structure will not give you any insights on the structure of the function being approximated.

This makes the analysis very hard, at least from the point of view of choosing right neural network architecture.

There is no "a standard and accepted method" for choosing network configuration.

Typically, as said previously, all neural networks have an *input* layer that do not perform any numerical processing and whose number of neurons is determined by the context since it is given by the number of features. For example, in this case it is the length of the arrays of autocorrelation values that we give in input.

Then the *output* layer has a number of nodes equal to the output variables that the neural network has to predict. In our case, since we aim to obtain the model paramaters starting from the autocorrelation function, if the parameters are 4 (for instance the

constants rate of the numeber of RNAs molecules increase and degradation and the same analogously for proteins in the simple first model analysed in chapter 2), the the output layer will have 4 nodes.

As regards the number of hidden layers and the relative number of nodes, as said before, there is no standard rule. We tried with a classical one that is the number of nodes decreases layer per layer.

Then we need to choose other main ingredients:

- one is the *activation function*. As the name suggests, it "activates" the output that the weighted sum of the inputs is transformed so that the output is limited into a determined range of values. In this case we use the *ReLU* function for each hidden layer and output layer. The rectified linear activation function (or ReLU) for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero (see Fig.6.3).

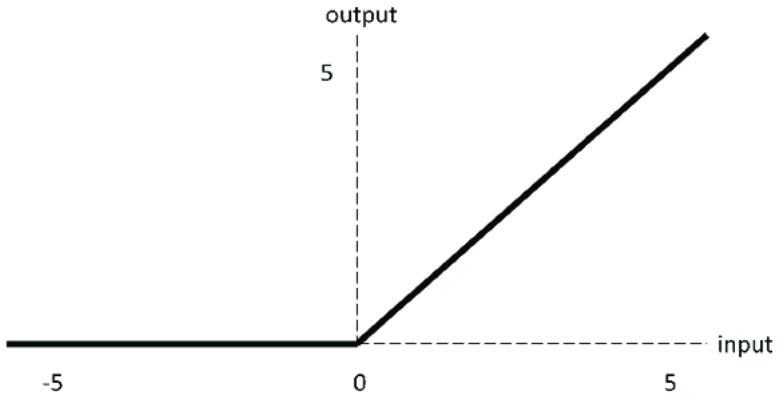

Figure 6.3: Sketch of ReLU activation function. Image from [76]

  Indeed we are going to predict values that are posive numbers (i.e. the model parameters) hence this is suitable for our context.

- One other ingredient is *the optimization algorithm*. The learning strategy of our neural network is based on backpropagation algorithm that updates all the weights (i.e. connections between neurons) in order to minimize *an error function* which characterizes the comparison between the network output and the ground truth in supervised models. The variation in time of the weights is given by:

$$\dot{w_i} = \frac{dE}{dw} \tag{6.3}$$

  where E is the error function that is also refered to as energy function in analogy to a physical system in which forces act to minimize the energy. There are dif-

ferent optimizers to make the network evolve, i.e. algorithms used to change the attributes of a neural network such as weights and learning rate in order to reach the optimal solution and minimize the loss. In this study we use the *Adam* as optimization algorithm that deals with the challenge of choosing the proper learning rate (in a classic gradient descent method, if it is too small leads to painfully slow convergence, if too large can hinder convergence) and of avoiding getting trapped in the suboptimal local minima. Its strategy is to accelarate the learning rate but it dampens oscillations towards local minimum. It combines the heuristics of two methods called Momentum and RMSProp.

- As regards the error function instead, since we are building a regressor model, we use the *Mean Absolute Error*. The mean absolute error of a model with respect to a test set is the mean of the absolute values of the individual prediction errors on over all instances in the test set. Each prediction error is the difference between the true value and the predicted value for the instance [77].

$$mae = \frac{\sum_{n=1}^{n} abs(y_i - \lambda(x_i)}{n} \tag{6.4}$$

where y_i is the true target value for test instance x_i, $\lambda(x_i)$ is the predicted target value for test instance x_i, and n is the number of test instances.

6.3 ANN hyperparameters

We must manually tune the following hyperparameters:

- the *batch size*: computing the gradient on the whole dataset is computationally very complex so nowadays the choice is to devide the dataset into batches, that is *the number of samples to work through before updating the internal model paramaters* (i.e. weights). If we set a batch size equal to 128, the model weigths will be updated after each batch of 128 samples.

- *number of epochs*: it defines the number of times that the learning algorithm will work through the entire training dataset. One epoch means that each sample in the training dataset has had an opportunity to update the internal model parameters.

- *dropout*: since there can be problems of overfitting so that the network learns the input data without being able to generalize or also the problem to get stuck in local minima of the energy function that implies a not optimal learning of the network, there are some regularization procedures to overcome these problems, one of which is perturbing the architecture of the network (dropout). This method consists in removing a subset of nodes during training, randomly choosing the removed units

at each training epoch. The reason is that it was observed that while training a network, after overfitting, the weights for some of neurons increases and cause the network to be dependant on them. By exploiting dropout, we are not dependant on any node anymore due to it is possible to drop it while training.

For example, if you set dropout rate to 0.1, then for each iteration within each epoch, each node in that layer has a 10% probability of being dropped from the neural network. This makes the network more robust and allows weights to explore a wider parameter space.

6.4 Coefficient of determination

Finally, we decide to judge the goodness of our model in predicting the true values using a score called *coefficient of determination* denoted as R^2 pronounced "R squared".

A data set has n values marked y_1,...,y_n each associated with a fitted (or modeled, or predicted) value f_1,...,f_n. Define residuals as $e_i = y_i - f_i$. If $\bar{y}$ is the mean of the observed data:

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{6.5}$$

then the variability of the data set can be measured with two *sums of squares* formulas: the sum of squares of residuals $SS_{res} = \sum_i (y_i - f_i)^2 = \sum_i e_i^2$ and the total sum of squares (proportional to the variance of the data) $SS_{tot} = \sum_i (y_i - \bar{y})^2$.

The most general definition of the coefficient of determination is:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{6.6}$$

In the best case, the modeled values exactly match the observed values, which results in $SS_{res} = 0$ and $R^2 = 1$. A baseline model, which always predicts $\bar{y}$, will have $R^2 = 0$. Models that have worse predictions than this baseline will have a negative R^2.

We are going to use this to evaluate the true vs predicted data. The better the linear regression fits such data, the closer is the value of R^2 is to 1.

Chapter 7

Regressor ANN at work

We generate the k parameters values that are distributed according to a Gamma distribution. This choice is due to the fact that we recognize these values as the most probable to be for the constant rates of reactions.

The probability density function for gamma is [78]:

$$f(x, a) = \frac{x^{a-1}e^{-x}}{\Gamma(a)} \tag{7.1}$$

for $x \geq 0$, $a \geq 0$. Here, $\Gamma(a)$ refers to the Gamma function. a is the shape parameter.

Gamma distributions are sometimes parameterized with two variables, with a probability density function of:

$$f(x, \alpha, \beta) = \frac{\beta^{\alpha}x^{\alpha-1}e^{-\beta x}}{\Gamma(\alpha)} \tag{7.2}$$

Note that this parameterization is equivalent to the above, with $scale = \frac{1}{\beta}$.

We choose to generate gamma distribution parameters with $a = 3$ and $scale = 0.5$ using scipy.stats function.

Finally, before starting, let us keep in mind that we generate 1000 autocorrelation data calculated with 5000 lags (relative to the SSA simulations time series results). Each set is refered to particular molecular species. Hence we consider three cases: the RNAS+Proteins autocorrelation case, the RNAs and proteins autocorrelations considered separately.

We decide to keep the model hyperparameters and the structure of the neural network the same from one genetic model to an other. Even though, the most important reason to choose this approach is to understand if neural networks can be an alternative method to the Bayesian Approximate Computation, keeping all fixed makes easier to compare which are the molecular species that carry more information about the model parameters.

In particular we decide to use a batch size equal to 128. This is a trade off between a low batch size that would make the simulation much slower and high batch size that

would require more memory storage and, finally, because it is the most suggested value to use in websites discussions. This last reason may seem a bit "superficial" but in this field experience from others help in the study. And we trained untill 50 epochs.

7.1 ANN applied to the first model

Let us consider the first genetic model described in section 2.2 with regulation that tends to keep the gene more active than inactive, i.e. $k_a = 1$ and $k_i = 0.5$.

In order to judge the performance of our regressor ANN, we plot we true parameters vs the predicted one, before evaluating the values of the R^2. We start from the autocorrelation of both RNAs and proteins given as input.

What we can notice is that test data (never seen by the neural network) are sparse but most of these are close or lie on the diagonal (Fig.7.1).

In this case $R^2_{test} = 0.5$ and $R^2_{train} = 0.9$. The score for the train is much higher than the one for the test. This maybe due to overfitting as suggested by the shape of the learning curve in Fig.7.3.

Since the test set is 5%, one attempt to get better performance was to increase the test set to 10%. This produces infact learning curves with a smaller gap between the training loss curve and the test loss curve (see Fig.7.4). The the scores in this case are $R^2_{test} = 0.4$ and $R^2_{train} = 0.7$.

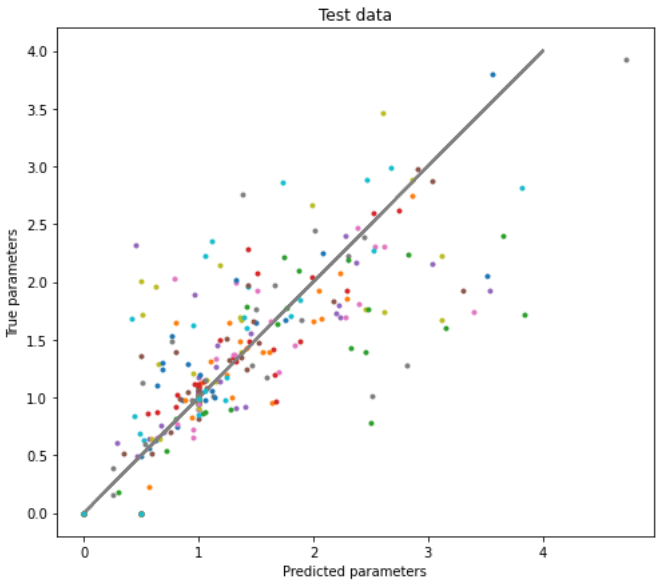

Figure 7.1: True data vs predicted data for the test set. RNAs and proteins autocorrelation are considered.

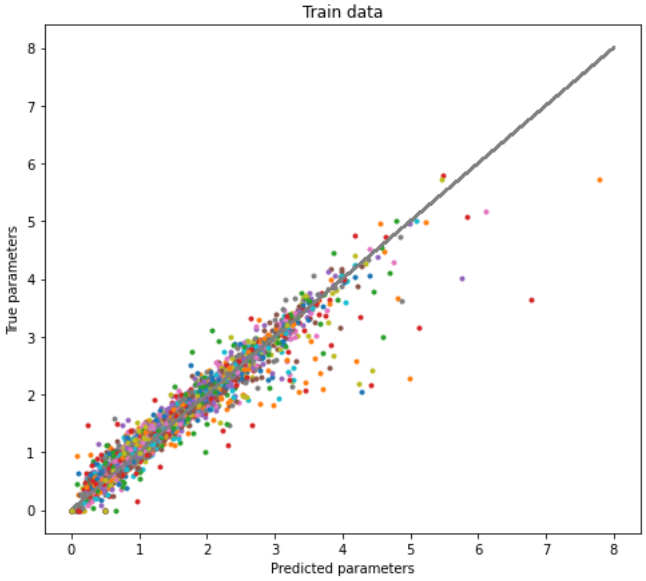

Figure 7.2: True data vs predicted data for the train set. RNAs and proteins autocorrelation are considered.

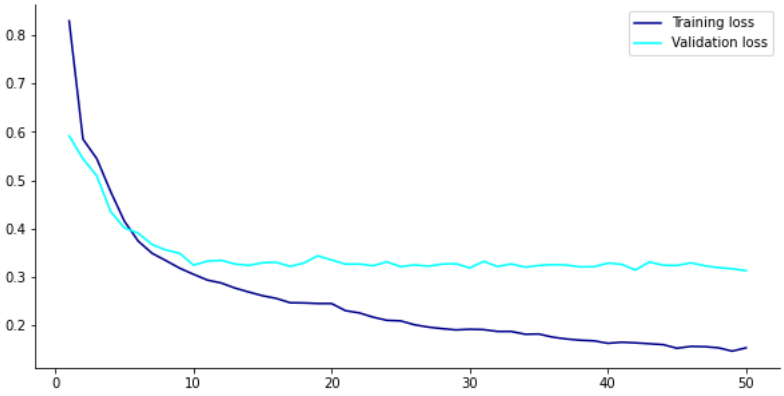

Figure 7.3: Learning curve of the ANN refered to RNA and proteins autocorrelations as input.

However the very interesting thing about this plots is that the neural network is learning. This gives hopes for its use in the model parameters estimation. Moreover, as regards the situation in which we consider only proteins autocorrelation given as input, the score for the test data is lower than the previous case, i.e. $R^2_{test} = 0.09$ and $R^2_{test} = 0.7$ suggesting that in this first genetic model, the RNAs and proteins autocorrelations contain a greater amount of information about the model parameters under investigation and this is as expected.

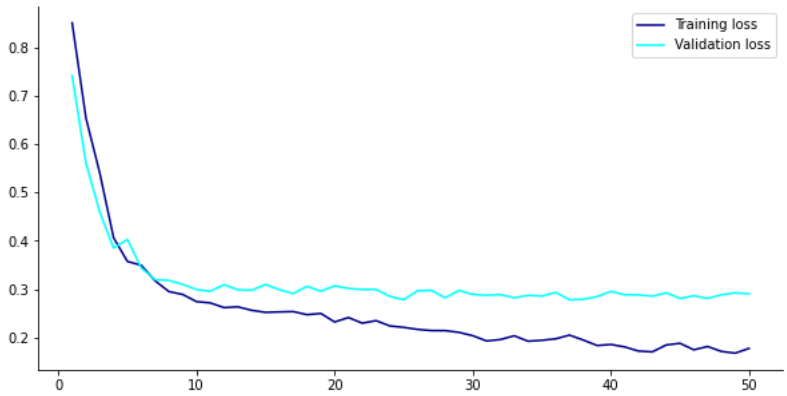

Figure 7.4: Learning curve of the ANN refered to RNA and proteins autocorrelations as input. The test set is 10%.

7.2 ANN applied to the autorepressor model

In the autoreprepressor model case, we start considering the values of the scores summerized in table 7.1 without annoying the reader with all the predicted vs true paramaters plots and learning curves that, anyway, are reported in Appendix.

	R^2_{test}	R^2_{train}
RNAs+Proteins	0.5	0.9
RNAs	0.2	0.6
Proteins	-0.06	0.4

Table 7.1: Autorepressor R^2 values for test data.

The surprise of this case is that proteins autocorrelation are less suitable for parameter estimation while RNAs carries more information. Finally as expected, the RNA+Proteins autocorrelation given as input to the regressor neural network helps most for parameter estimation.

7.3 ANN applied to the toggle-switch model

Even in this case we report the table 7.2 of the R^2 values. Since this system is made of two genes, each one producing their number of RNA and protein molecules, we decided to consider the autocorrelation relative to the RNAs molecules of both genes and the same is for proteins.

In this case, RNAs + proteins autocorrelation values given as input is not reported because of lack of computational memory in order to do the simulation. However, as in

	R^2_{test}	R^2_{train}
RNAs	0.1	0.2
Proteins	0.1	0.2

Table 7.2: Toggle switch R^2 values for test and train data.

the first model and autorepressor case, we should expect that they carry more information about the distribution of model parameters.

Here, the interesting aspect that arises is that R^2 values of RNAs molecules are equal to the proteins molecules hence in a system like this using RNAs autocorrelation or proteins autocorrelation as input to the neural network would give the same amount of information about the model parameters (at least using the neural network with the hyperparameters chosen by us).

Anyway, leaving the biological discoveries apart, in all these genetic models, we can notice that the network is learning, specially from the relative loss curves (all reported in Appendix C) and R^2 values. This opens the way to an alternative method for the paramters distribution estimation.

Part IV

NF-κB, a stochastic active player in human cancer

Chapter 8

The NF-κB family of transcription factors

Nuclear factor kappa-light-chain-enhancer of activated B cells (abbreviated as NF-κB) was originally identified as a family of transcription factors that bind the enhancer of the immunoglobulin κ light-chain gene, whose function is to ensure the expression and secretion of functional antibodies and contribute to antigen binding by increasing the variability of the antibodies [32]. This was discovered by Ranjan Sen and his collegues in their experiments that identified a protein binding to a specific, conserved DNA sequence in nuclei of activated B lymphocytes. They named it for the cell type in which they identified it and the gene it affected: Nuclear Factor binding near the κ light chain gene in B cells [58]. Although its function in the regulation of immunoglobulin light-chain gene remains unclear, NF-κB plays critical roles in development, survival, and activation of B lymphocytes [31]. So it has been considered key regulator of antibodies production.

However, NF-κB has been found in all cell types implicated in the transcriptional events controlling multiple cellular outputs. It comprises a family of structurally-related eukaryotic transcription factors, which bind to consensus DNA sequences at promoter regions of responsive genes regulating a large number of normal cellular and organismal processes such as immune and inflammatory responses, developmental processes, cellular growth and apoptosis and, for such reason, its name contains information not completely correct since it is neither a fundamental regulator of κ light-chain gene nor B-cell specific.

Nowadays, NF-κB is actually considered as a pleiotropic mediator of gene expression control.

There are five proteins in the mammalian NF-κB family including NF-κB1 (also named p50), NF-κB2 (also named p52), RelA (also named p65), RelB and c-Rel. All proteins of the NF-κB family share a *Rel homology domain* (RHD). The RHD is composed of two structural domains: the N-terminal *DNA binding domain* and the C-terminal domain that has an immunoglobulin-like fold that functions as a *dimerisation domain* [10]. Hence, RHD is required for binding to spiecific DNA binding sites in the promoters of

target genes and serves as a dimerization interface to other NF-κB transcription factors since these members combine to form different dimers that regulate the transcription of target genes.

To date, the NF-κB signaling system consists of about a dozen different dimers composed by a different combinations of five homologous protein.

Only p65, c-Rel and RelB have a C-terminal *transactivation domain* (TADs), which is responsible for the initiation of gene expression, while p50 and p52, lacking TADs, can *positively regulate transcription* (i.e. increases the expression of a gene) through heterodimerization with TAD-containing NF-κB family members or other protein that have a trans-activating capability.

Alternatively, p50-p52 homodimers can act as *negative regulators of gene transcription* (i.e. they bind to silencers, thus inhibiting the expression of the gene) by competition with TAD-containing dimers for DNA binding.

Thus, homodimers constitutive binding to κB sites may require a replacement by transcriptionally competent dimers, enforcing an activation threshold for certain target genes.

8.1 The IκBs proteins: master regulator of NF-κB signaling

In unstimulated cells, NF-κB proteins are sequestered in the cytoplasm through binding to molecules that are the IκBs family. There are height known IκBs members: IκBα, IκBβ, IκBϵ, IκBζ (NFKBZ), BCL-3 (B-cell lymphoma 3) and the precursors p100 (NF-κB2) and p105(NF-κB1).

All IκBs members share an *ankyrin domain* which is crucial for specific interaction with the Rel-homology domain. Ankyrin repeat, one of the most widely existing protein motifs in nature, they are found in all three kingdoms of life and consist of 30-34 amino acid residues and exclusively functions to mediate protein-protein interactions in order to activate or suppress biological processes [33]. Some of these interactions are directly involved in the development of human cancer and other diseases [34].

Below we sum up the most important characteristics of each one of these members:

- the most known member of the family is **IκBα**. p50/p65, that is one of the major dimers involved in the canonical pathway (NF-κB pathways will be described in the next section 8.2), is mainly bound to IκBα.

  Upon signal stimulation, IκBα is phosphorylated in the N-terminal domain and degraded thus allowing dimers to translocate into the nucleus to activate gene transcription.

  The activation of NF-κB causes the transcriptional upregulation of IκBα which in

turn represents an autoregulatory negative feedback loop, responsible for the shut off the signal.

Thus, IκBα is the primary product of *a rapid and transient induction of NF-κB activity.* However, in presence of a persistent stimulus, NF-κB is maintained in the nucleus despite the upregulation of IκBα mRNA synthesis, and this persistent activation is regulated by IκBβ.

- Like IκBα, also **IκBβ** is phosphorylated but it *undergoes a slower degradation followed by a re-synthesis and accumulation in the nucleus as hypophosphorylated form, which can bind the NF-κB dimers.*

  DNA-bound NF-κB:IκBβ complexes are resistant to newly synthesized IκBα suggesting that hypophosphorylated, nuclear IκBβ may prolong the expression of certain genes.

- The most recent member to be described is **IκBϵ** that is degraded after phosphorylation similarly to IκBα, but its degradation and resynthesis are delayed compared to those of IκBα.

  These differences in the kinetics of degradation have relevant effects on the nature of the transcriptional response to stimuli.

  IκBϵ is primarly expressed in hematopoietic cells (i.e. immature cells that can develop into all types of blood cells, including white blood cells, red blood cells, and platelets and are found in the peripheral blood and the bone marrow [35]) and its loss results in selective defects in hematopoietic lineages, although it appears that IκBϵ loss is largely compensated by IκBα.

  Moreover, IκBϵ is differentially expressed during B-cell development and has been proposed to regulate both p65- and c-Rel-containing NF-κB complexes in B cells.

- On the contrary, **IκBζ** and **BCL-3** (B-cell lymphoma 3) are different regulators of NF-κB signaling.

  BCL-3 is the unique member of IκB family that contains a TAD; it is found in the nucleus associated with p50- and p52-containing homo- and heterodimers.

  BCL-3 may mediate *the release of transcriptional repression by removing p50 homodimers from κB sites*, thus mediating activation by acting on repressive NF-κB dimers and allowing p65:p50 or other TAD-containing dimers access to DNA.

  Alternatively, BCL-3 may also *stabilize repressive p50 homodimers and thus inhibit NF-κB activation* or, in the case of p52 homodimers, it can confer transcriptional potential in an inducible manner.

IκBζ is an other unusual member of IκB family which is more similar to BCL-3 than the rest of the family. *It is not expressed constitutively but rather is upregulated in response to particular stimuli.*

Following NF-κB activation, IκBζ is expressed and then it associates primarily with p50 homodimers.

Furthermore, IκBζ is found associated with p50 on the promoter of IL-6, which is not expressed in IκBζ knockout cells, and it is, therefore, hypotized that IκBζ acts as a coactivator for p50 homodimers.

IκBζ is also been reported to negatively regulate p65-containing NF-κB- complexes, as demonstrated by the slight elevation of NF-κB activity observed in IκBζ knockouts.

- Finally the less known family member is **IκBγ**, a 70 kDa molecule detected only in lymphoid cells.

  Its sequence is identical to the IκB like C-terminal region of p105; indeed IκBγ is the product of an alternative promoter usage that produces an mRNA encoding the C-terminal portion of p105.

  Although, initially, it was thought that IκBγ functioned as a trans-inhibitor of Rel-proteins, similar to the other IκB proteins, subsequent studies have suggested that IκBγ probably inhibits only p50 or p52 homodimers.

8.2 NF-κB signaling pathways

The release of NF-κB dimers from its inhibitors is mediated by **IKK** complex. Upon stimulus, IKK is transformed from its neutral form (IKKn) into its active form (IKKa), a form capable of phosphorylating IκBα, leading to IκBα degradation.

The two mains components are IKKα and IKKβ, serine/threonine kinases which are characterised by the presence of an N-terminal *kinase domain*, a C-terminal *helix-loop-helix* (HLH) *domain* and a *leucine zipper domain.*

The IKK complex phosphorylates IkBα on N-terminal serines, and this triggers its degradative polyubiquitination through the proteasome (a simple sketch of this process is shown in figure 8.1).

Although both IKKα and IKKβ are capable of phosphorylating IκBα and IκBβ, they are less efficient kinase for IκBβ than for IκBα, and this difference may explain the delayed degradation kinetics of IκBβ following stimulation with Tumour Necrosis Factor (TNF)α. This difference in degradation rate is in fact one of the characteristics of this IκB member that was already mentioned in the previous section 8.1.

IKKα and IKKβ dimerization are dependent on the *leucine zipper domain*, which is therefore required for kinase activity.

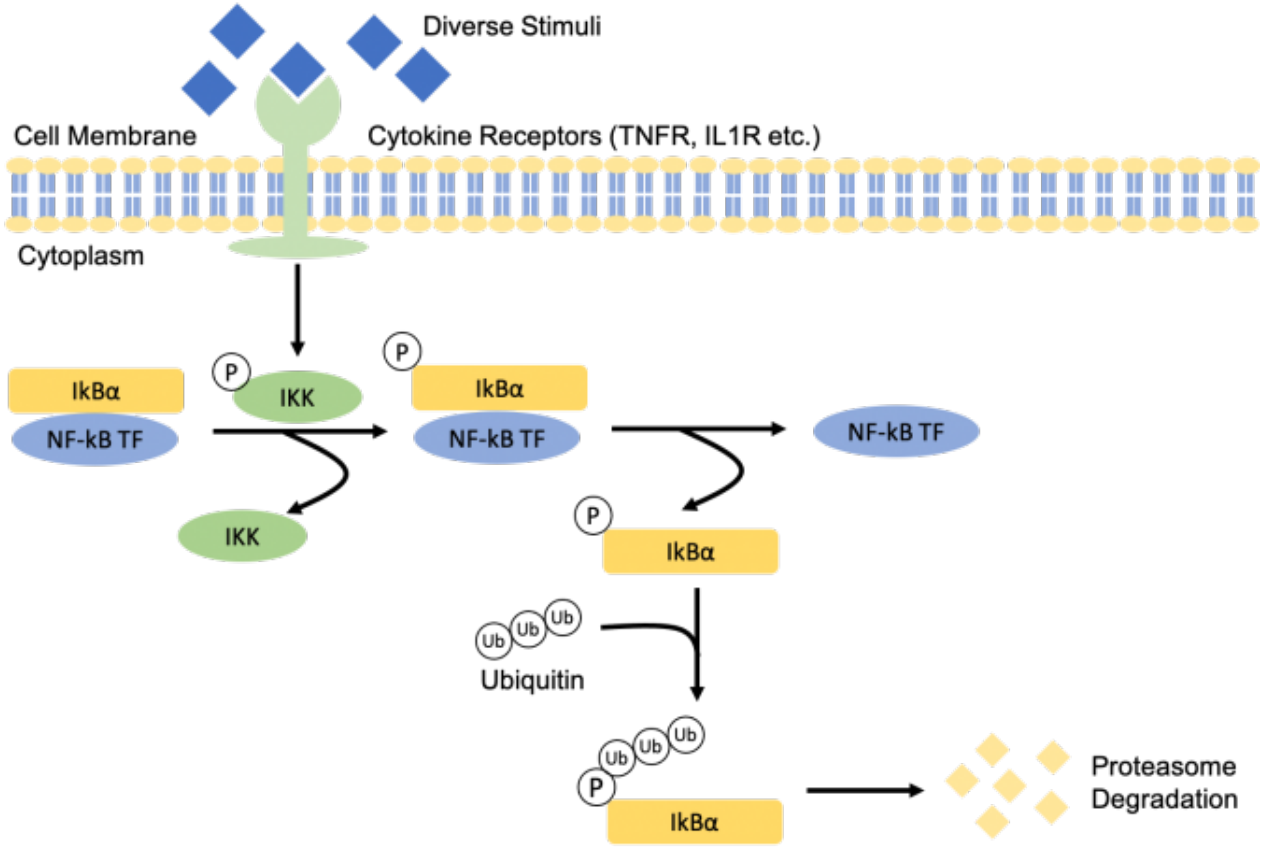

Figure 8.1: The binding of pro-inflammatory cytokines to its receptors triggers the activation of IKK which will phosphorylate IkBα releasing the sequestered NF-κB Transcription Factor. The IkBα will then undergo proteasome degradation to synthesize new proteins. Image from [14].

Together with these two catalytic subunits, there is the regulatory subunit *NF-kB essential modifier* (**NEMO**-also known as IKKγ) which is not related to IKKα and IKKβ and is characterized by a *C-terminal zinc-finger like domain*, a *leucine zipper*, *N-terminal and C-terminal coiled-coil domains*.

Both IKKα and IKKβ interact with NEMO through a C-terminal hexapeptide sequence termed the *NEMO binding domain* (NBD).

Although in vitro assembly of the complex indicates that only IKKβ assembles with NEMO, data from IKKα knockout mice indicate instead that also IKKα/NEMO complexes are readily formed.

From here, the NF-κB signaling pathways have been classified into two distinct and evolutionary conserved types, according to the two multiprotein IKK complexes which regulate the stimulus-responsive phosphorylation and consequent degradation of IκB

proteins: the classical or so-called "canonical" NF-κB signaling pathway where IKKβ is sufficient for IκBα and IκBβ phosphorylation and degradation, while IKKα may regulate gene expression in the nucleus by modifying the phosphorylation status of the histones.

On the contrary, in the alternative or "non-canonical" pathway IKK complex consists exclusively of an IKKα homodimer, which leads to the phosphorylation of p100 and its consequent processing to p52.

Even though both classic and alternative pathways are principally activated during the functioning of the immune system, the first is mainly involved *in inflammation response* as in the regulation of proliferation and apoptosis of lymphoid cells throughout the immune reaction; the latter regulates the lymphoid organogenesis and so its engangement occurs mainly after non-inflammatory stimuli.

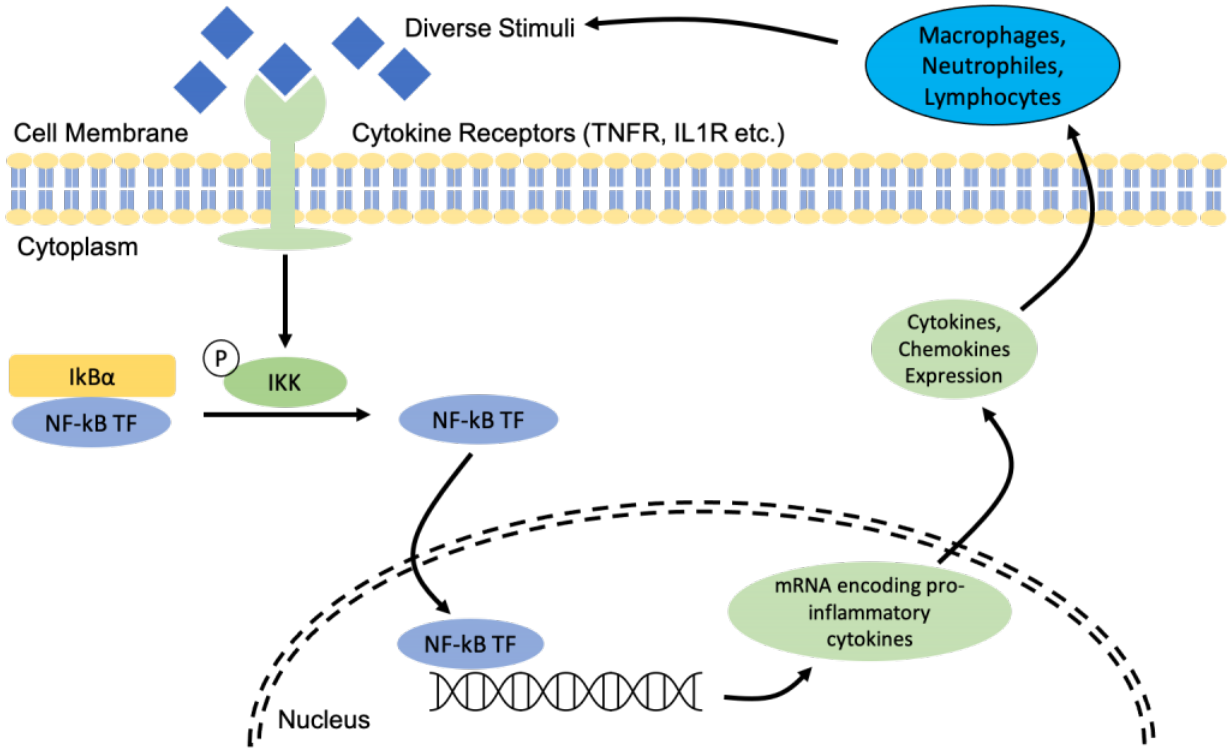

Figure 8.2: Activation of canonical pathway during inflammation related to obesity. NF-κB will induce transcription expression of pro-inflammatory cytokines when bound to DNA. Pro-inflammatory cytokines will recruit immune cells to the site. Presence of immune cells further produces pro-inflammatory cytokines which could bind to the cytokine receptors that are responsible for activating NF-κB. Image from [14].

In the canonical pathway, after pathogen-derived lipopolysaccharide (LPS) mediates inflammatory stimulus and cytokines such as TNF and interleukin (IL)-1, , the IκB proteins IκBα, IκBβ and IκBϵ are phosphorylated by the "canonical" IKK complex on two specific N-terminal serines.

The subsequent ubiquitination mediated by the ubiquitin ligase β-TRCP makes IκB available for proteasomal degradation, so the NF-κB dimers are capable to translocate in the nucleus and activate gene transcription. This triggers transcription of the inhibitors and numerous other genes. Among the synthesized inhibitors, there are IκBα and A20. The newly synthesized IκBα again inhibits NF-κB, while A20 inhibits IKK by catalysing its transformation into another inactive form (IKKi), in which it is no longer capable of phosphorylating IκBα.

Usually, NF-κB activity can be detected within ten minutes after stimulation and some NF-κB responsive promoters are induced almost immediately. As already mentioned, the activation of NF-κB causes the transcriptional upregulation of IκBα which causes *a powerful negative feedback mechanism* that is responsible for the NF-κB repression, whose activity can be restored after an other stimulus and may result in an oscillatory pattern of NF-κB activity during chronic stimulation.

Conversely, multiple non-inflammatory stimuli are responsible for the engagement of the "non-canonical" pathway in a variety of cell types.

In B cells, the survival factor B-cell-activating factor belonging to the TNF family (BAFF) and, in splenic stromal cells, lymphotoxin β signaling are able to activate NF-kB for many hours or days.

Although the activation mechanism remains partly unclear, it is thought that the precursor protein p100, which dimerizes with RelB, is partially processed to p52 upon IKKα activation by NF-κB inducing kinase (NIK).

However, p100 processing was shown to be a co-translational mechanism, related to continuing p100 synthesis. Thus, p52 DNA-binding activity seems delayed in comparison with p50/p65 complexes.

It is interesting to note that the non-canonical activation of NF-κB is slower than canonical one, and it lacks a negative feedback control because the RelB-containing dimers are not subject to the high dynamic regulation of the IκB proteins.

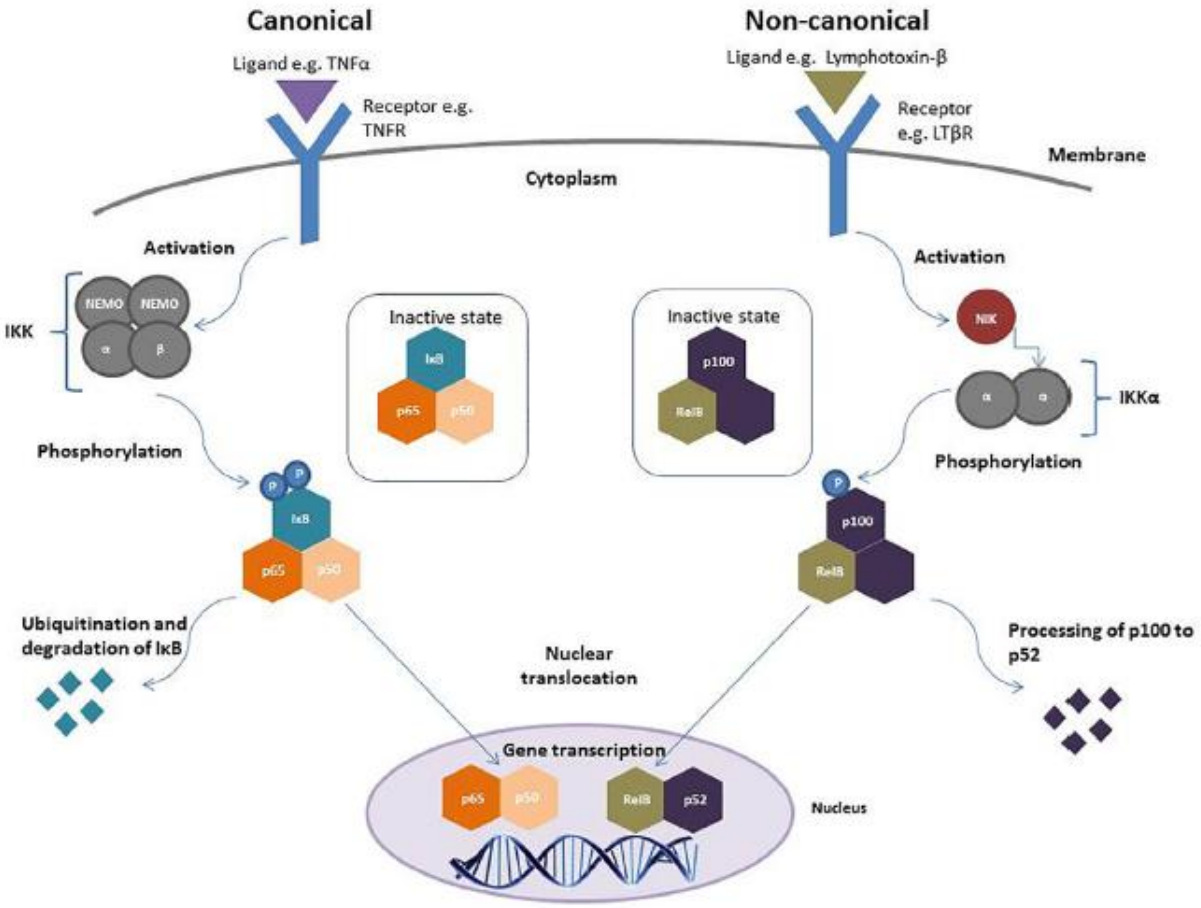

Figure 8.3: Sketch of canonical vs non-canonical NF-κB signaling pathway (Patel et al.,2018)

8.3 Models of NF-κB activity oscillations

Many mathematical models of such key regulator NF-κB activity oscillation have already been studied during years, highilighting the fact that it is strongly regulated by the inhibitor protein IκBα. Here we list some studies results:

- For the first time, a mathematical model was presented by Hoffman et al. Their research showed that the resulting negative feedback produces a propensity for oscillation in NF-κB activity.

  They analysed the IκB-NF-κB signaling module using knockout cell lines for the three IκB isoforms (IκBα,IκBβ and IκBϵ) in order to evaluate the contribution of each one on the temporal control of NF-κB oscillation.

It was shown that the coordinated degradation, synthesis and localization of all three IκB isoforms is required to generate the characteristic NF-κB activation profile.

The model displayed that IκBα is responsible for strong negative feedback and a rapid switch-off of the NF-κB response, while IκBβ and $-\epsilon$ explained their action dampening the system's oscillatory potential and making stable the NF-κB responses during longer stimulations.

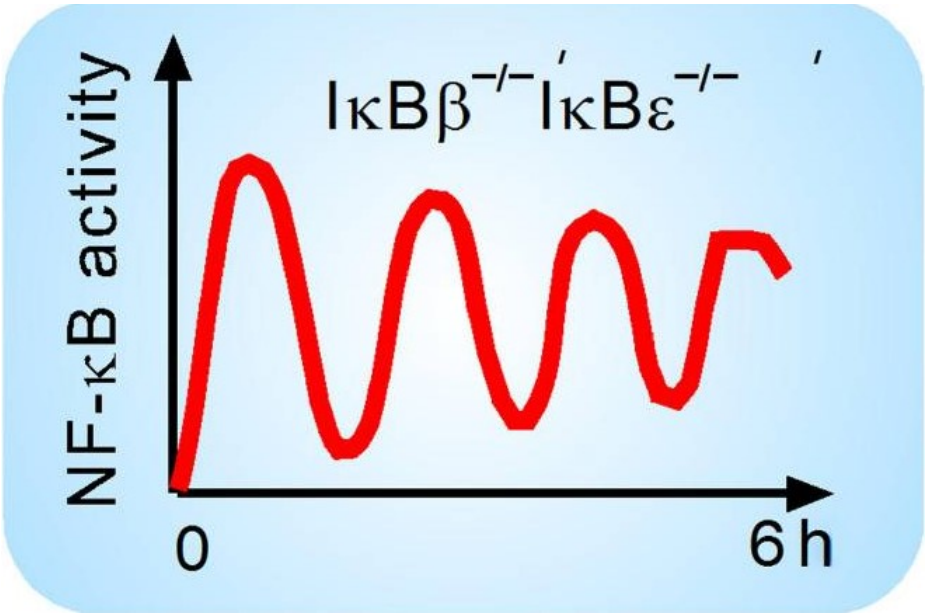

Figure 8.4: Schematic of oscillatory time course of NF-κB in response to persistent TNFα in IκBβ and IκBϵ knockout cells. Image from [40].

Moreover, the researchers identified a *bimodal temporal IκB-NF-κB signaling behaviour depending on stimulus duration that is shown to generate also specificity in gene expression.*

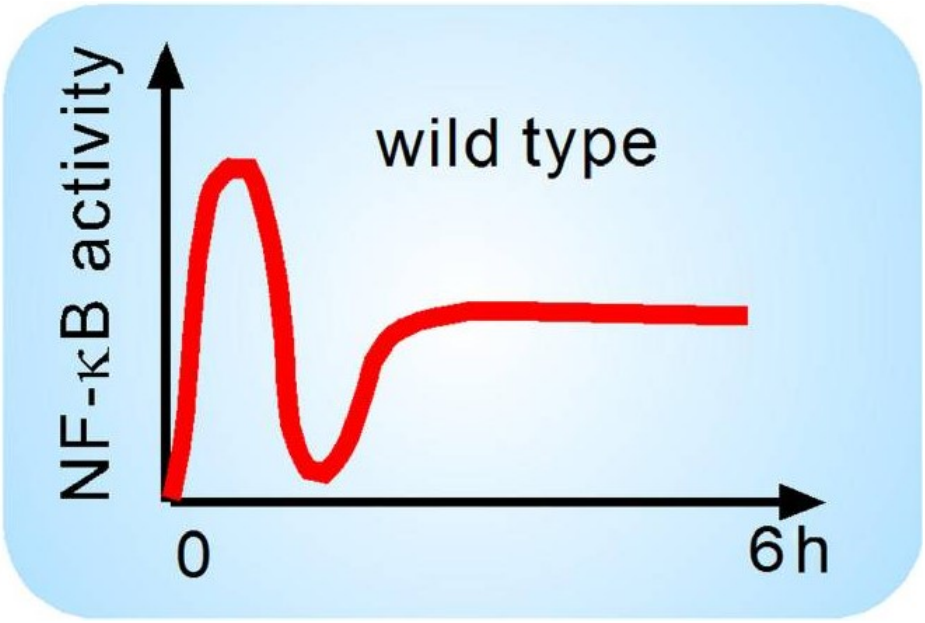

Figure 8.5: Characteristic biphasic time course of NF-κB signaling in response to TNFα in various wild-type cells. NF-κB activity peaks around 30 min, drops to basal levels around 1 h, and rises to an intermediate level thereafter [40].

- Based on this model, Kearns et al. explained better IκBϵ role, whose NF-κB-induced transcription is delayed relative to that of IκBα.

  IκBϵ delayed synthesis provided an antiphase negative feedback that effectively dampened IκBα mediated oscillations.

  Furthermore, they demonstrated that both these negative feedback regulators are necessary for the termination of NF-κB activity and NF-κB mediated gene expression in response to transient stimuli.

- According to Mothes et al. and Wang et al., the oscillation was explained in detail through a bifurcation analysis; the models had two steady states, stable focus and unstable focus, with a stable limit cycle, depending on such bifurcation parameters which represented the intensity of the TNFα stimulation, total NF-κB and IκBα transcription rate.

- Inoue et al. mathematically analysed the NF-κB oscillation by using the core model based on three components, IKKβ, IκBα and NF-κB. According to their stability analysis, the NF-κB activity showed two characteristic behaviours, *oscillation and switch-like activation*, regulated by positive and negative feedback, respectively.

  Switch-like activation was fundamental to compensate for the upstream IKK tran-

sient activity and to induce accumulation of nuclear NF-κB, which in turn was responsible for subsequent oscillation in target gene expression.

Thus, the NF-κB oscillation was merely a damped oscillation.

However, the NF-κB activity is neither a damped nor sustained oscillator phenomenon but is *a stable transient event, in which different players are involved, such as RelA*.

The transcription of the NF-κB target genes depends on the phosphorylation of both IκBα and RelA subunit of NF-κB, but no mathematical model considering RelA phosphorylation have been previously proposed.

So Hatanaka et.al constructed a model in which the focus was the influence of phosphorylation of IκBα and RelA on the nuclear-cytoplasmic oscillation.

They confirmed that NF-κB signaling module transiently responded to the concentration of active IKKβ in a dependent manner. However, when the IKKβ concentration was fixed, the amplitude of the oscillation became larger as the value of total NF-κB increased, suggesting that the oscillation period was regulated also by RelA phosphorylation.

Therefore, controlling the phosphorylation process, it may be possible to govern the NF-κB oscillation driving the NF-κB dependent expression gene.

8.4 NF-κB and colorectal cancer

As said at the beginning, NF-κB proteins are involved in the control of a large number of normal cellular and organismal processes. However, on the other hand, these transcription factors are persistently active in a number of disease states, including cancer, arthritis, chronic inflammation, asthma, neurodegenerative diseases, and heart disease [9]. NF-κB activation supports tumorigenesis by enhancing cell proliferation and angiogenesis, inhibiting apoptosis, and promoting cell invasion and metastasis.

No triggering mutations of NF-κB in CRC have been described; however, constitutive activation of NF-κB has been reported and is associated with higher tumor stages, treatment resistance and poor survival outcomes.

The role of NF-κB in adenoma formation and development of colitis-associated cancer has been studied for the first time in a mouse model, where deletion of IKKβ in intestinal epithelial cell (IEC) and myeloid cells had distinct outcomes; deletion of IKKβ in IECs showed reduced adenoma incidence, which had a direct effect in tumor progression, as demonstrated by reduction in antiapoptotic B-cell lymphoma 2 (Bcl-2) protein, Bcl-x_L . On the contrary, deletion of IKKβ in myeloid cells was associated with a less reduction in adenoma but showed reduced expression of genes encoding proinflammatory cytokines. So, the IKKβ-mediated NF-βB activity has a cell-type specific role in the early-stage of carcinoma formation demostrating that the NF-κB functions are cell-type and tumor-type specific.

However, no known studies have mapped NF-κB expression in different human tissue concerning adenoma-carcinoma transformation. Furthers evidence demonstrate crosstalk between NF-κB pathways and various other signaling networks implicated in CRC progression.

For example, growth factors can trigger the activation of the IKK complex through signaling pathways described in figure 8.6.

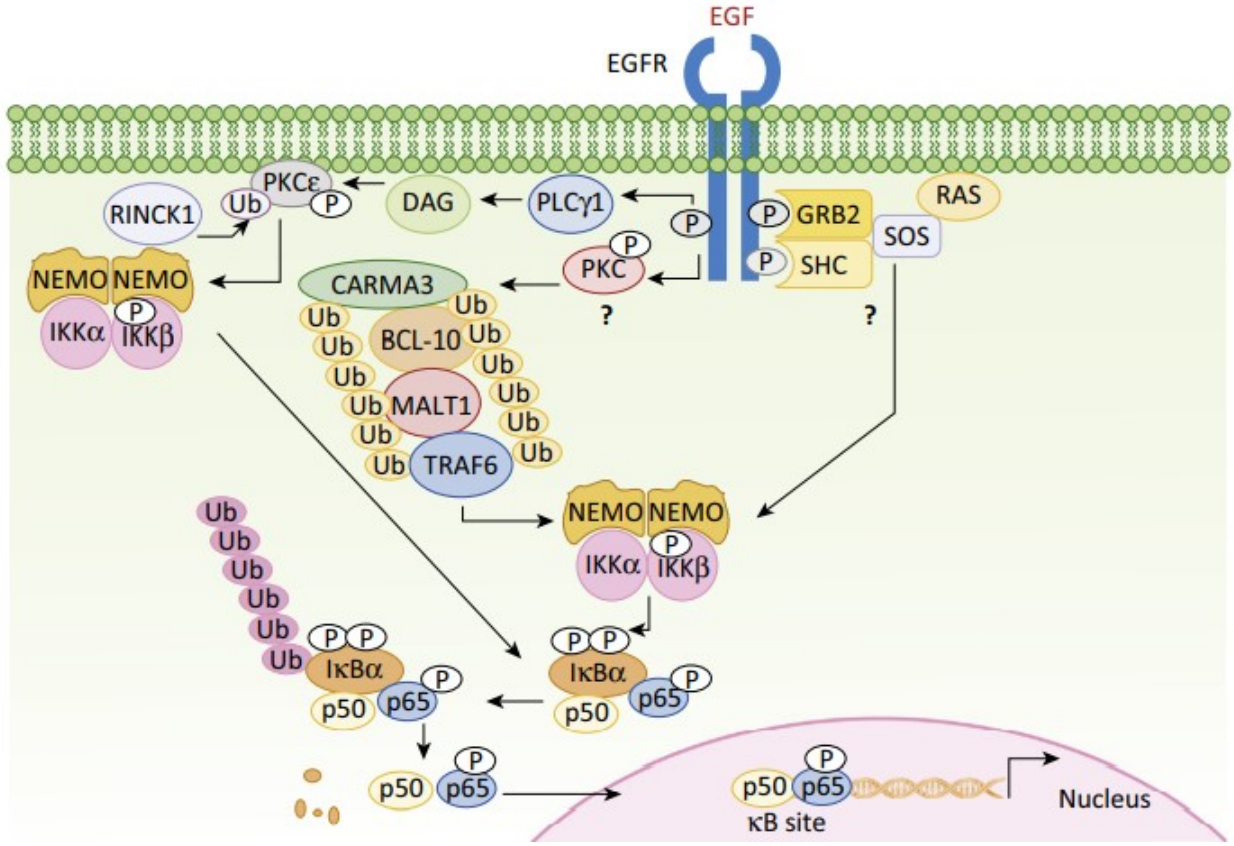

Figure 8.6: Molecular mechanisms by which EGFR activates NF-kB. Image from [15].

Growth factor is a substance made by the body that functions to regulate cell division and cell survival [19]. Epidermal growth factor (EGF) is a protein that stimulates epidermal and epithelial cell growth and differentiation by binding to its receptor called Epidermal Growth Factor Receptor (EGFR) [20]. The epidermal growth factor receptor is a transmembrane protein that is involved in cell signaling pathways that control cell division and survival [18]. Although it is now well established that EGF activates NF-κB through the IKK complex, signaling molecules that link EGFR activation to the IKK complex have only been recently characterized.

In particular, constitutive epidermal growth factor receptor and nuclear factor κb

activities are seen in multiple solid tumors and combine to provide oncogenic signals to cancer cells. Recent studies have also defined mechanisms by which resistance occurs through crosstalk between EGFR- and NF-κB-dependent pathways in colon cancer cells.

Chapter 9

NF-κB model formulation

Let us consider the very common heterodimer p50-p65 in the canonical pathway. This NF-κB dimer as well as the IKK complex will be considered as single proteins and so the dynamics leading to the formation of these complexes will be disregarded. Hence we do not consider second order reactions. The regulatory system considered has two activators IKK and NF-κB and two *inhibitors* A20 and IκBα. Indeed, the nuclear activity of NF-κB is terminated by the newly synthesized IκBα, which enters the nucleus, binds to NF-κB and takes it out into cytoplasm; A20 instead acts as IκBα activator and so an NF-κB inhibitor.

In the presence of an extracellular signal such as TNF or IL-1, IKK is transformed into its active (phosphorylated) form.

In this form it is capable of phophorylating IκBα which in turn leads to its degradation. As a result, degradation of IκBα releases the second activator, NF-κB.

The free NF-κB enters the nucleus and upregulates transcription of the two inhibitors IκBα and A20 and of a large number of other genes, in particular the target gene whose gene expression has to be regulated by NF-κB.

The newly synthesized IκBα again inhibits NF-κB, while A20 inhibits IKK by catalysing its transformation into another inactive form, in which it is no longer capable of phosphorylating IκBα. We ignore this and we consider that the protein A20 inhibits IKK making it neutral and so we do not consider the step of its transformation into IKKi (that is a catalytic reaction because A20 catalyses its transformation into its inactive form) and the kinetics of its resynthesis to its neutral form IKKn.

And this is the logic of our first model: we consider that the four players IKK, NF-κB, IκBα and A20 can be either active or inactive. In other words, they can be either ON or OFF.

Whereas, the RNA produced starting from the target gene of the NF-κB transcription factor can be modeled using ordinary differential equation in deterministic framework. Using the notation of the Chapter 2, given m the number of RNA molecules produced by the target gene and k_4 the rate of RNA production, k_5 the rate of degradation, we

can write an analogous equation to Eq.2.5:

$$\frac{dm}{dt} = k_4 \cdot g(\cdot) - k_5 m \tag{9.1}$$

where the promoter activity function $g(\cdot)$ is determined by the NF-κB state.

Then a simple sketch of the model is shown in Fig.9.1:

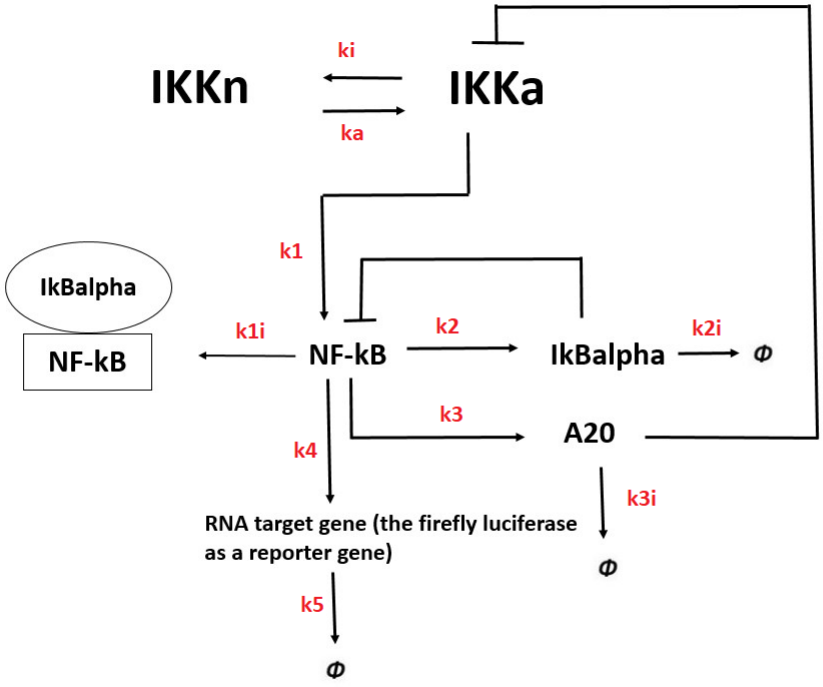

Figure 9.1: Sketch of our first model NF-κB feedback loop regulatory module with relative parameters.

There are two repressing actions caused by the inhibitors and from the schemes we can see this very well: one caused by IκBα directly on NF-κB and one mediated by A20 on IKK in its active form.

In this simple model there are already 10 rates constant:

- ki, the constant of activation rate of IKK.
- ka, the constant of inactivation rate of IKK.
- k1, the constant of activation rate of NF-κB.
- k1i, the constant of inactivation rate of NF-κB (this process is shown in Fig.9.1 as the NF-κB attached to its inhibitor IκBα).

- k2, the constant of activation rate of Ikα.
- k2i, the constant of degradation rate of Ikα.
- k3, the constant of activation rate of A20.
- k3i, the constant of degradation rate of A20.
- k4, the constant of RNA production of target gene that, in experimental terms, is the firefly luciferase as reporter gene in the experiment made in University of Bologna where Cignal Lenti Reporter System (QUIAGEN) was used.
- k4, the constant rate of RNA degradation.

Hence the rates of activation and inactivation of the four ON/OFF mentioned players are the following. Starting from the ones that are "repressed", i.e. IKK and NF-κB:

- $IKK_{activation} = ka(IKKn)(\frac{1}{1+A20_{active}})$
- $IKK_{deactivation} = ki(IKKa)(A20_{active})$
- $NFkB_{activation} = k1(NFkB : IkB\alpha)(IKKa)(\frac{1}{1+IkB\alpha})$
- $NFkB_{deactivation} = k1i(NFkB)(IkB\alpha)$

Whereas for the inhibitors:

- $IkB\alpha_{activation} = k2(NFkB)(IkB\alpha_{inactive})$
- $IkB\alpha_{deactivation} = k2i(IKKa)(A20_{active})$
- $A20_{activation} = k3(A20_{inactive})(NFkB)$
- $A20_{deactivation} = k3i(A20_{active})(NFkB)$

Where IKKn, IKKa, $A20_{active}$, $A20_{inactive}$, NFkB, $I\kappa B\alpha$ can be either 0 or 1. $NFkB : IkB\alpha$ is the NF-κB bound to its inhibitor and also this can be either 0 or 1 denoting the fact that in this model NF-κB can be in its active or inactive form.

These are the rates to consider when simulating this model using Gillespie's algorithm.

This model is very simple is just a "first order kinetic reactions model" of the NF-α feedback loops. However notice that the approximation of the ON/OFF behavior for what regards the NF-κB is not so "abstractive" if we think at the dynamics of a ligand binding to a macromolecule such as protein that is described by the Hill function [74]. Protein-ligand binding typically changes the structure of the target protein, thereby changing its function in a cell. In this case the binding of IκBα to the NF-κB "changes the NF-κB from an ON state to a OFF state".

Part V

Results and discussion

Chapter 10

Results and discussion about the NF-kB model simulation

The expectation from the simple NF-κB model described in the last section 9 in the model formulation is to obtain an oscillating RNA molecules temporal series as in the model predicted by Hoffman et al. where they experimentally measured the NF-κB activity in the IκBα and IκBϵ knock-out cells, whose trend was reported in section 8.3 in Fig.8.4. As said, the model displayed that IκBα is responsible for strong negative feedback and a rapid switch-off of the NF-κB response.

Fig.10.1 shows the trend of the SSA simulation of the number of RNA molecules produced from the target gene considering the simple NF-κB model (whose scheme was reported in Fig.9.1). The parameters used are chosen with the logic to keep the NF-κB more active than inactive and are: $ka = 1$, $ki = 0.1$, $k1 = 1$, $k1i = 0.1$, $k4 = 1$, $k5 = 0.1$, $k2 = 0.1$, $k2i = 1$, $k3 = 0.1$, $k3i = 1$. Moreover we start from a configuration where everything is inactive: $IKKa = 0$, $IKKn = 1$, NF-κB is inactive so $NF - \kappa B = 0$, IκBα is inactive so $I\kappa B\alpha = 0$ and finally $A20 = 0$.

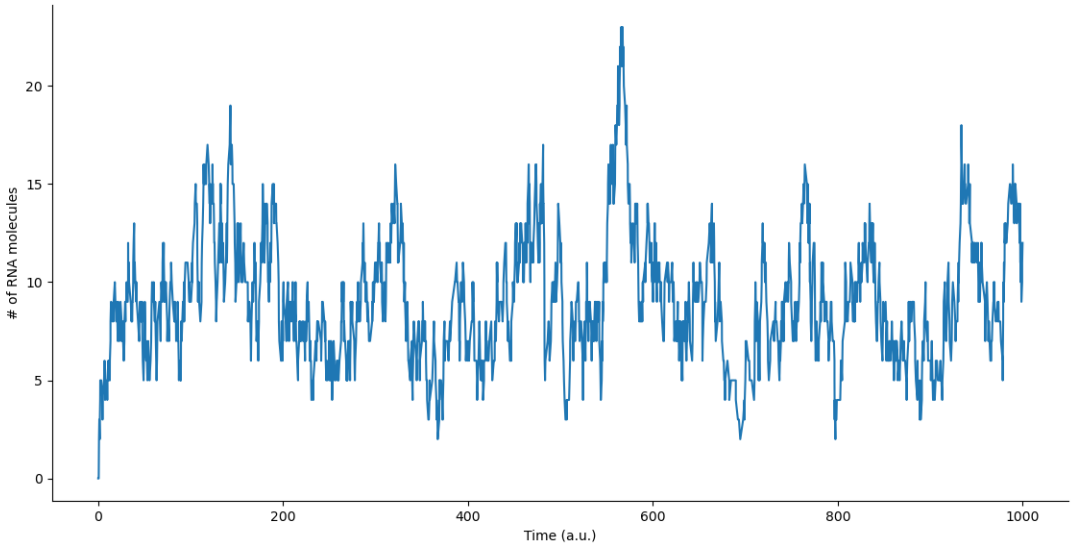

Figure 10.1: Number of RNA molecules produced through the NF-κB activation and inhibition simulated by using the Stochastic Simulation Algorithm and considering model in Fig.9.1

At first sight, we find an oscillatory profile whose "degree" of oscillation can be verified calculating the autocorrelation as we did for our first genetic models.

It has an oscillatory behavior as well as the 4 autocorrelation curves reported in Fig.10.3 for 4 SSA simulations of the same model.

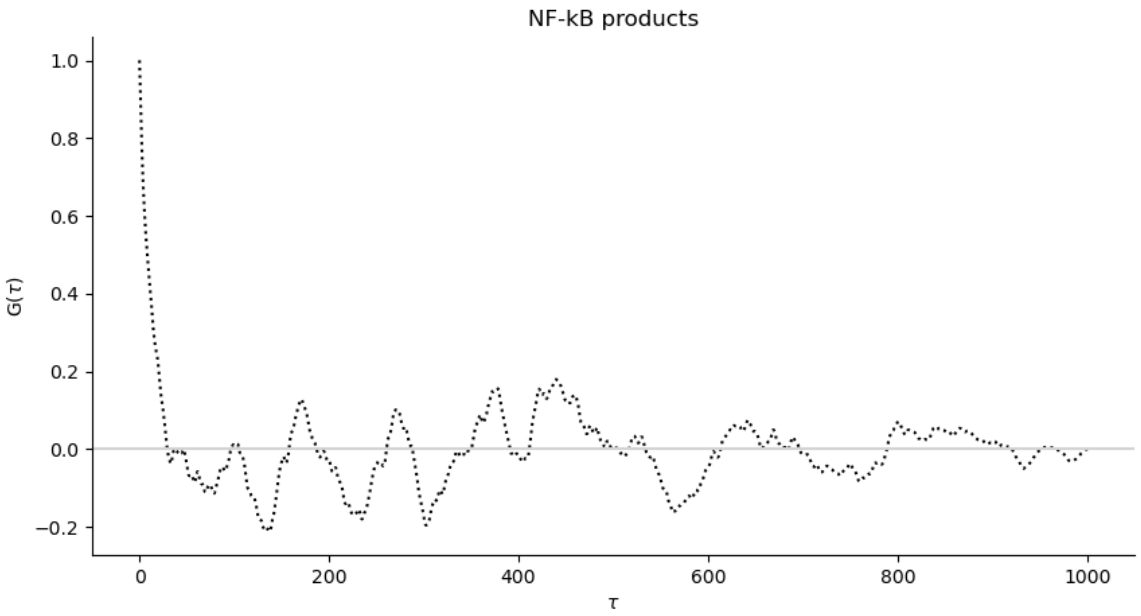

Figure 10.2: Autocorrelation curve regarding the temporal series in Fig.10.1. Number of lags equal to 100000 and sampling period dt=0.01. The y-axis reports the autocorrelation values and on the x-axis is the sampling period.

What we can notice is that they remind us to the toggle switch behavior where each simulation has a particular oscillatory behavior. It is not like the autorepressor case where there is a common initial deep characterizing all simulations (refering to the same model with same parameters but using a different random seed). Hence the autocorrelation calculated using quantiles (Fig.10.4) shows large error bars from the second point (the first one has zero error since the autocorrelation is 1 for all the simulations) and then from a sampling point equal to around 600 it decreases as confirmed from Fig.10.3 where we see a common behavior.

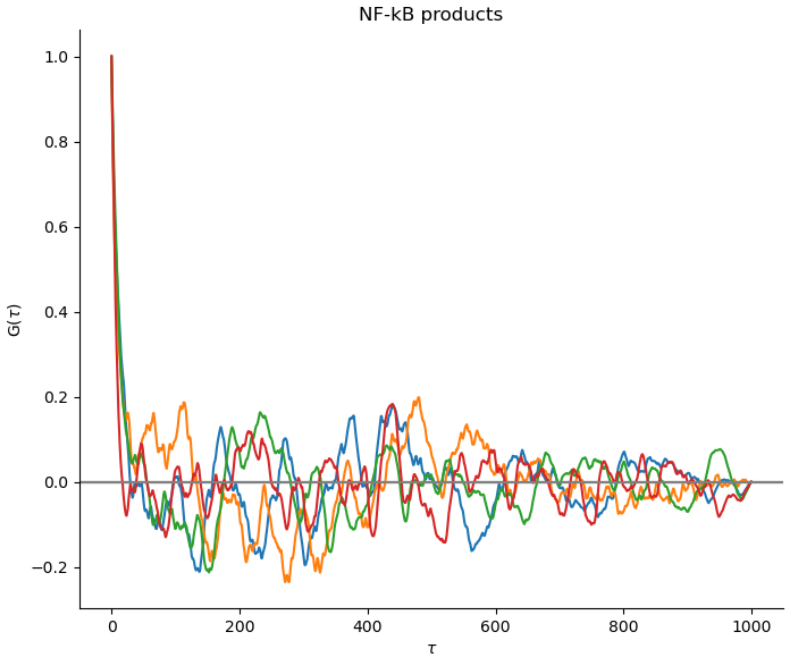

Figure 10.3: Autocorrelation curves regarding 4 SSA simulations of the same model 9.1 with the same parameters. Number of lags equal to 10000 and sampling period dt=10. The y-axis reports the autocorrelation values and on the x-axis is the sampling period.

We can start to "put our model under test" by switching off one of the most studied inhibitor, namely IκBα. Hence we consider the model constant rates previously describe but with $k2 = 0$, $k2i = 0$ when simulating the IκBα knock-out. We report the plot of only the autocorrelation calculated with quantiles using 64 simulations in Fig.10.5.

We can notice that the medians points lie much more on the zero horizontal line and this is a sign of a lower oscillating signal and thus a more constituive expression behavior. In Appendix D the time trend of a SSA simulation made using this parameter is shown together with the case in which we put also $k3 = 0$, $k3i = 0$ (see respectively Fig.D.1 and Fig.D.2). A brief comment about them: one can notice an oscillating behavior at the beginning and then there is a dampening. This is what we see by eye looking at the time series but the autocorrelation computed with quantiles for the case in which only k2 and k2i are equal to zero shows a general less oscillating behavior, as just said.

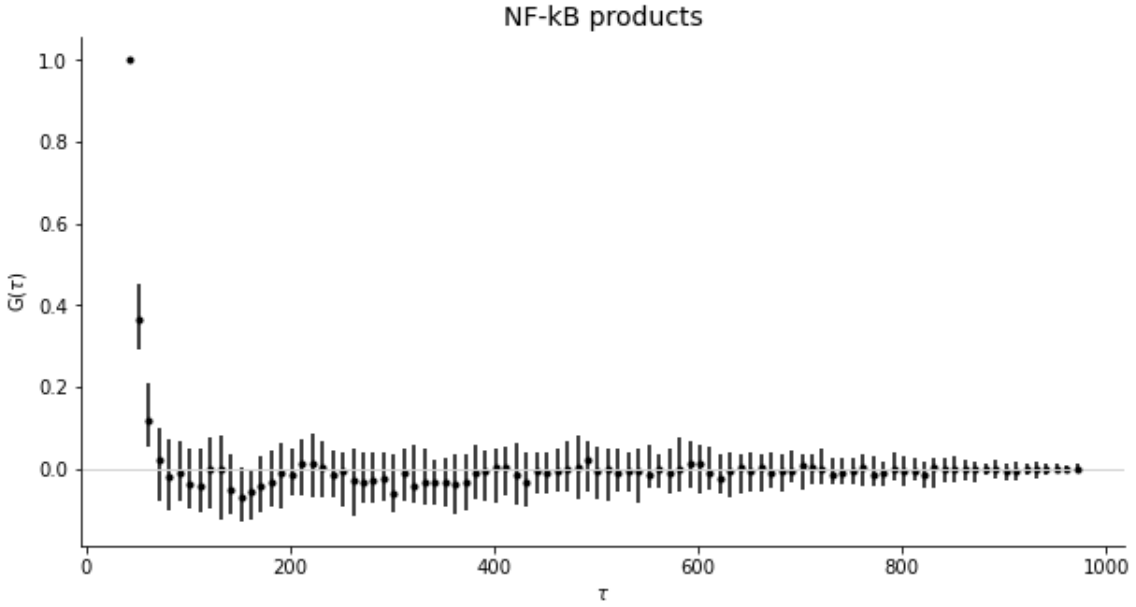

Figure 10.4: Autocorrelation curve plotted calculating quantiles of 64 simulations at each time step. Number of lags equal to 10000 and sampling period dt=10. The y-axis reports the autocorrelation values and on the x-axis is the sampling period.

Now coming back to the original model with all constant rates paramters different from zero, as in the case of the "standard" genetic models analysed, from the Fourier Transform analysis (Fig.10.6) we can notice relevant peaks at low frequencies and the noise is evenly sparse towards higher frequencies suggesting the presence of a periodicity also in this case.

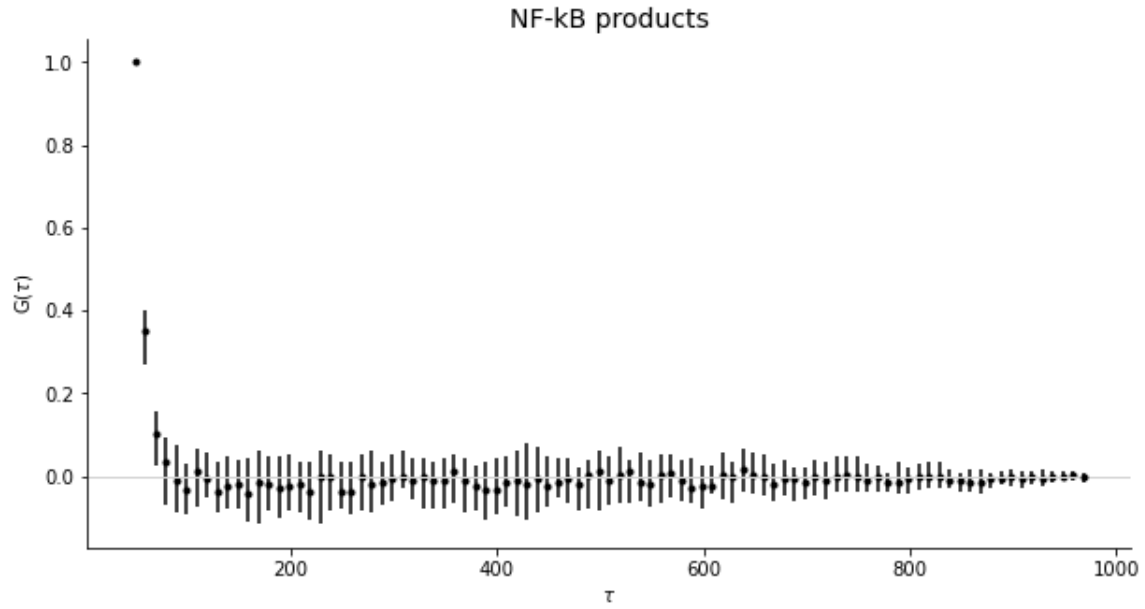

Figure 10.5: Autocorrelation curve plotted calculating quantiles of 64 simulations at each time step. Number of lags equal to 10000 and sampling period dt=10. The y-axis reports the autocorrelation values and on the x-axis is the sampling period. In this case $k2 = 0$, $k2i = 0$.

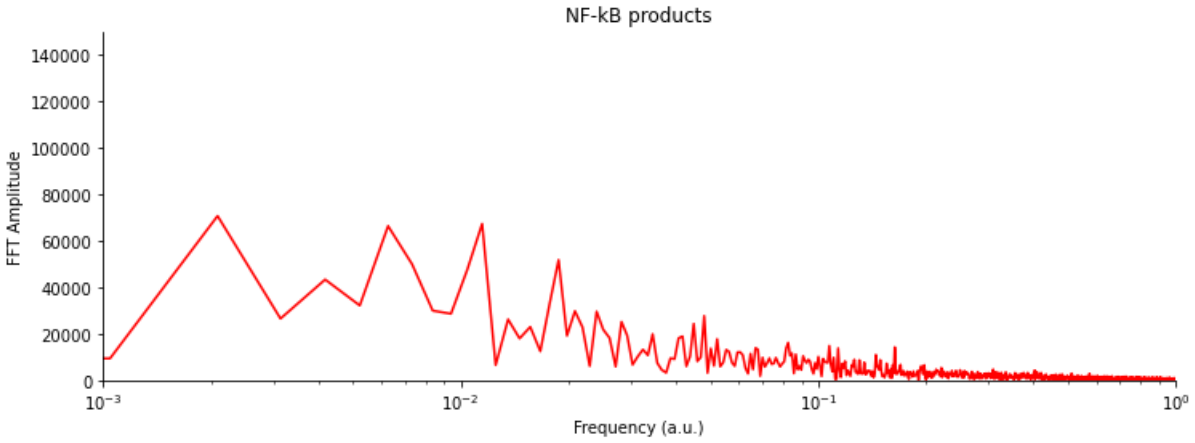

Figure 10.6: Fast Fourier Transform of the temporal signal in Fig.10.1. y-axis is in linear scale while the the x-axis is in log scale.

For concluding the pattern recognition analysis of our simple NF-kB model activity, the continous wavelet transform is shown in Fig.10.7.

The first impression is that it is similar to the CWT of the constituive expression

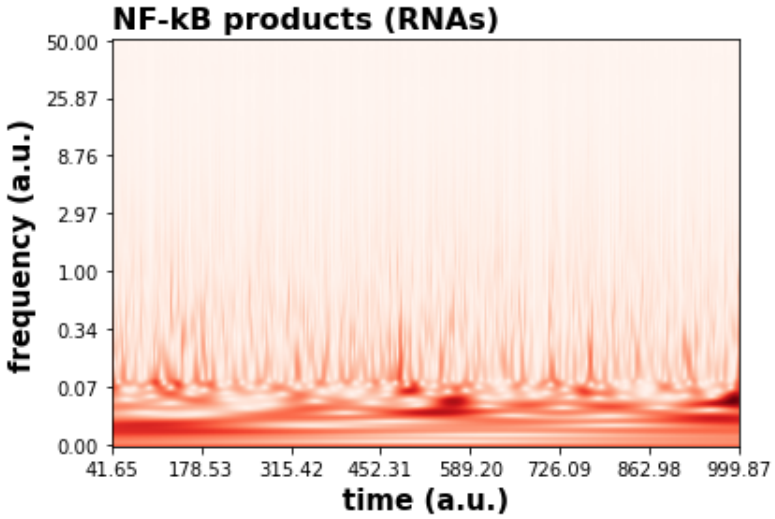

Figure 10.7: Continous Wavelet Transform of the temporal signal in Fig.10.1 using Generalized Morlet Wavelt. Frequencies are reported on the vertical axis while the horizontal axis represents the time.

model (Fig.5.32 and Fig.5.30).

However if we observe it, we can notice two dark red spots at about 600 and 1000 arbitrary units of time and other regular patterns for example some "lighter openings" between them. All located at low frequencies as exprected from the FFT analysis.

It would be interesting to apply the CWT to a longer time of simulation but as explained in the section 5.4.3 of CWT application, it goes beyond the computational resources available.

Chapter 11

Conclusions

All this thesis work is triggered by the desire to give a mathematical modeling of the NF-κB activity, that has a fundamental role in the regulation of many genes but also its disfunction leads to a long list of illnesses such as the colonrectal cancer.

In this voyage towards the a first mathematical model, we developed a "hybrid" algorithm (SSA combined with tauleaping) that is much faster than the SSA when the number of molecules produced tends to be higher, but on the other side it looses its speed performance when we have for example a simple system made by a gene that tends to be more inactive than active and so producing a low number of RNA molecules. If we think about it, it is the same behavior of a simple tauleaping algorithm except the important difference that the hybrid one controls that the number of molecules do not become negative, which is unphysical. Moreover this is an "adaptive" algorithm namely SSA or tauleaping are applied on the basis of what the dynamics of the system under study are (that for instance maybe the case of the NF-κB). However, we did not feel to use it in this work of thesis because it needs to be tested on many other models together with other unit testings. Finally about this, one other improvement that should be done is the automatation of the choice about the threashold of differences between the old rates and the updated rates. By experience, fixing a $\Delta K_t = 10$ we achieve the performance just explained but we admit that in any case the choice of a parameter is not confortable.

Playing with the gene regulation of the first protein synthesis model has helped us in the building of the algorithm. Hence we exploited biology to build the hybrid algorithm.

Then we passed to the inverse procedure: by means of algorithms built for the study of stochastic processes we exploited the behavior of four simple standard genetic circuits models: the constituive expression, the regulated gene, the autorepressor and the toggle switch.

In particular, we exploited the autocorrelation, the fast fourier transform analysis and the continous wavelet transform.

What we are able to conclude about these simple models using these means of investigation is that all models analysed have inner periodicity (confirmed by the fast fourier

transforms results) but the autorepressor model is the "most periodic" one. First its autocorrelation curve is the only one that with computing quantiles of 64 simulations preserves the characteristic initial negative deep; then from the CWT analysis, we found patterns located at regular time intervals.

The periodicity in time of the signal in this context is very important. It could be responsible of phenotypic processes such as the example of synchronization between the myocardial contractile response and gene expression of contractile protein. Or even one can find the cause of this periodicity and discover important relationships that if broken could lead to cellular disfunctions. And one can act in order to repristinate this periodicity. Indeed, for each model we have searched also "their place in biology" and we found that one gene that uses this type of regulation is the Regulatoy factor X1. Many studies show that RFX1's gene expression regulation is based on its autorepression in response to DNA replication arrest. Also its oncogenic potential is analysed in some cancers.

However, the subject of this thesis if the NF-κB and also this is an other transcriptional factor and these reasonings can be done also in this context.

We ended this journey leaving a simple model of the NF-κB activity based on the only actions of two inhibitors, the IκBα and the A20. The results of this very simple model based only on first order reaction rate and ON/OFF behavior show a temporal series that has an oscillating behavior very similar to the one found by Hoffman et al. in their experiments. The autocorrelation function is indeed oscillating whereas the fast fourier tranform shows periodicity at low frequencies such as the first simple models analysed. From the continous wavelet transform some patterns arise but with the computational resources at hand it is difficult to draw conclusions.

The model lacks of many features such as catalytical reactions and so second order reactions. However, just in the introduction of this thesis we mentioned the famous phrase "all models are wrong but some are usefull". Our model is wrong but it is usefull as a starting point and also to predict what happens if one of the few players involved lacks. For instance the important role played by the inhibitor IκBα whose lacking leads to a constituive expression of NF-κB, that is what happens in colon cancer cells.

And, also in this case as mentioned in the introduction, once the model agrees with the experimental data all that remains is finding the model parametrs, an hope for finding the cure for illness.

The neural networks analysis has shown to work, given the autocorrelation functions as input, in the three simple genetic models considered at the begging and not only this. It reveals also the amount of information stored in each chemical species. For instance in the autorepressor case, our neural network works better when the autocorrelations given as input are those of RNA molecules and less for proteins.

Appendix A

Software used in this thesis

All simulation methods and statistical data analysis have been performed using Python 3.9 and the programs are available in the following GitHub page `https://github.com/ManuelaCarriero/PyExTra` where you can find a detailed explanation of all the Python codes and how to reproduce the results reported in this work.

Appendix B

Glossary of biology

- **Oncogenes** are genes that have the potential to cause cancer. In tumor cells, these genes are often mutated, or expressed at high levels [59].
- **Lymphoid cells** are a family of immune effector cells that have important roles in host defense, metabolic homeostasis and tissue repair but can also contribute to inflammatory diseases such as asthma and colitis [37].
- **Lymphocyte** is a type of cell produced by lymphoid tissue that passes into the blood. It is one of the three elements of the white blood series and mediates the immune response by recognizing foreign molecules (antigens) with its membrane receptors; two main groups of lymphocytes are distinguished: the l. B which are responsible for the production of antibodies and the l. T which govern the destruction of cells infected by viruses and bacteria, foreign cells and tumor cells.
- **B lymphocyte** is type of white blood cell that makes antibodies. Also known as B cells, they are part of the immune system and develop from stem cells in the bone marrow [39].
- A **knockout**, as related to genomics, refers to the use of genetic engineering to inactivate or remove one or more specific genes from an organism. Scientists create knockout organisms to study the impact of removing a gene from an organism, which often allows them to then learn something about that gene's function [36].
- **Interleukins** (IL) are a type of *cytokine* (i.e. are a broad and loose category of small proteins $\sim 5-20kDa$ important in cell signaling) first thought to be expressed by leukocytes alone but have later been found to be produced by many other body cells. They play essential roles in the activation and differentiation of immune cells, as well as proliferation, maturation, migration, and adhesion. They also have pro-inflammatory and anti-inflammatory properties. The primary function of interleukins is, therefore, to modulate growth, differentiation, and activation

during inflammatory and immune responses. Interleukins consist of a large group of proteins that can elicit many reactions in cells and tissues by binding to high-affinity receptors in cell surfaces. Interleukin 6 (**IL-6**) is an interleukin that acts as both a pro-inflammatory cytokine and an anti-inflammatory myokine (i.e. one of several hundred cytokines or other small proteins $\sim 5 - 20kDa$ that are produced and released by muscle fibers in response to muscular contractions). In humans, it is encoded by the IL6 gene.

- **Trans-inhibitor** is the catalytic inactivation of a homodimer, each of which subunits is catalytically active [38].

- **Tumor Necrosis Factor** is a protein made by white blood cells in response to an *antigen* (substance that causes the immune system to make a specific immune response) or infection. Tumor necrosis factor can also be made in the laboratory. It may boost a person's immune response, and also may cause necrosis (cell death) of some types of tumor cells. It is a type of *cytokine*. Also called TNF [70].

- **Phosphorylation** is a process in which a phosphate group is added to a molecule, such as a sugar or a protein [71]. Because phosphate groups are highly negatively charged, phosphorylation of a protein alters its charge, which can then alter the conformation of the protein and ultimately its functional activity [72].

Appendix C

Artificial Neural Networks plots

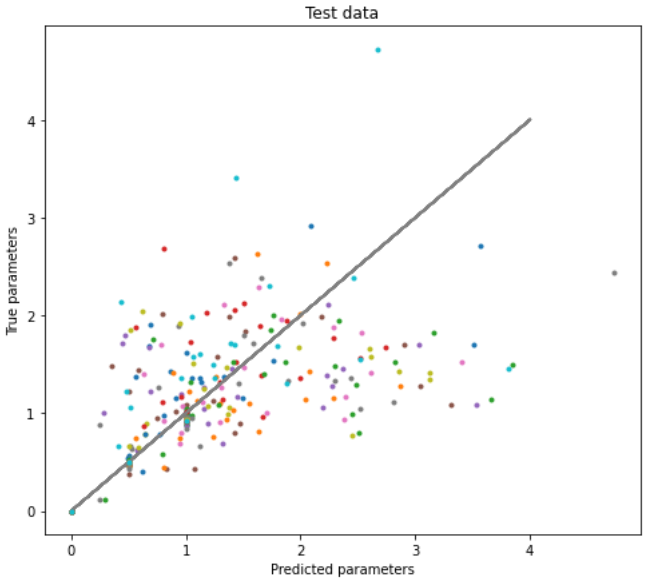

Figure C.1: First model true data vs predicted data for the test set. Only proteins autocorrelation are considered as input.

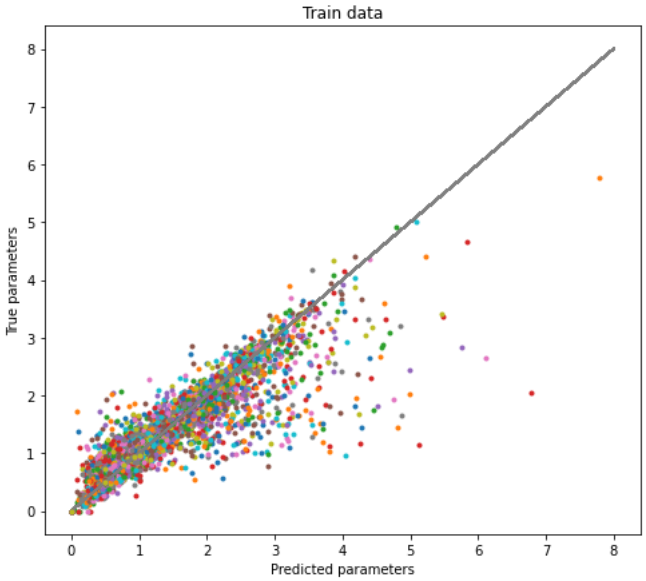

Figure C.2: First model true data vs predicted data for the train set. Only proteins autocorrelation are considered as input.

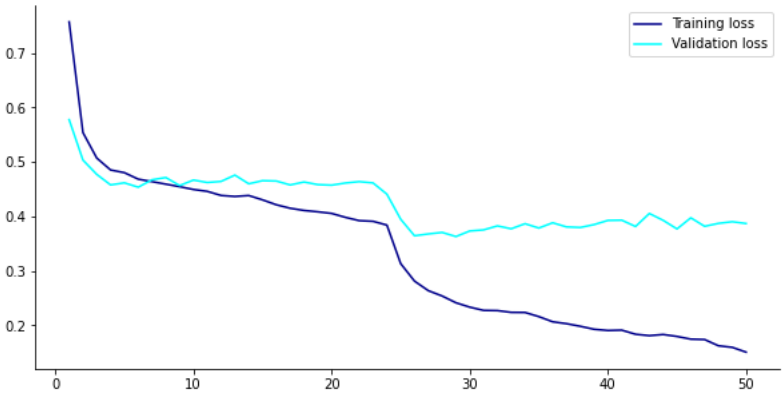

Figure C.3: First model learning curve of the ANN refered to only proteins autocorrelations as input.

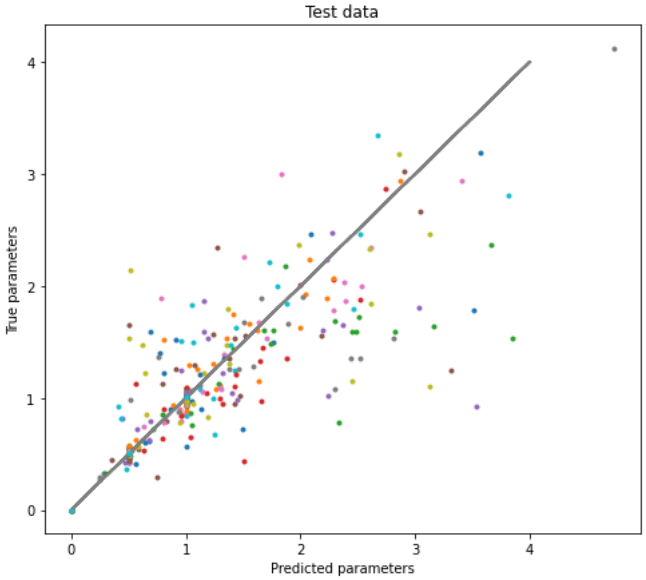

Figure C.4: True data vs predicted data for the test set in case of autorepressor model. RNA+PROTEINS autocorrelations are considered as input.

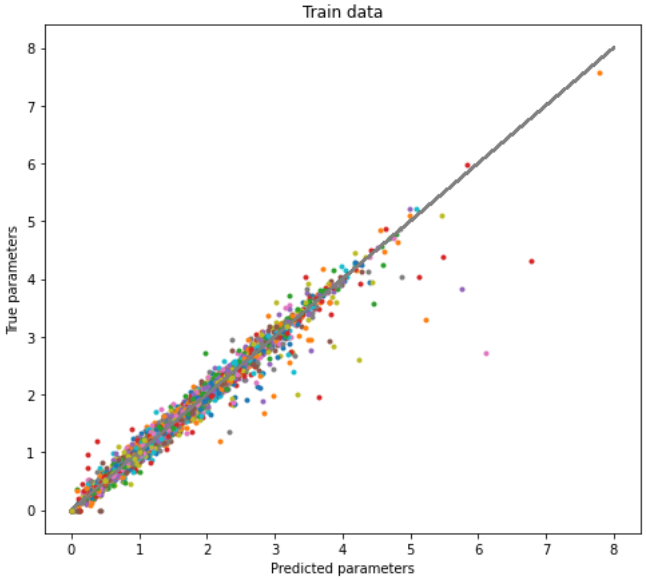

Figure C.5: True data vs predicted data for the train set in case of autorepressor model. RNA+PROTEINS autocorrelations are considered as input.

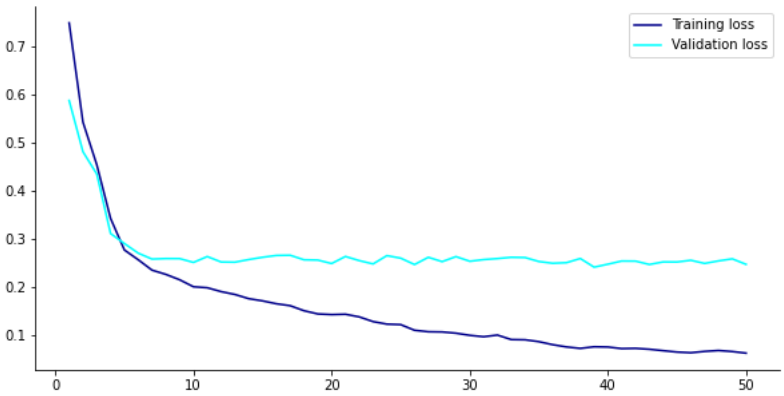

Figure C.6: Learning curve of the ANN refered to only proteins autocorrelations as input for the autorepressor model.

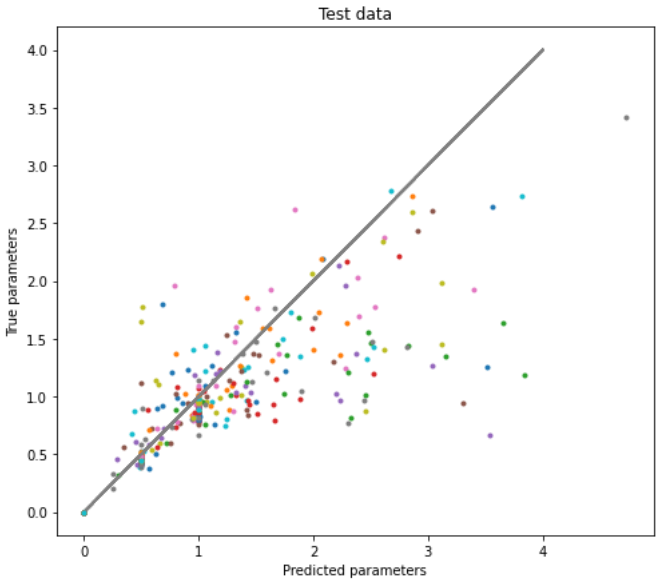

Figure C.7: True data vs predicted data for the test set in case of autorepressor model. RNA+PROTEINS autocorrelations are considered as input. Dropout is added in the Neural Network model.

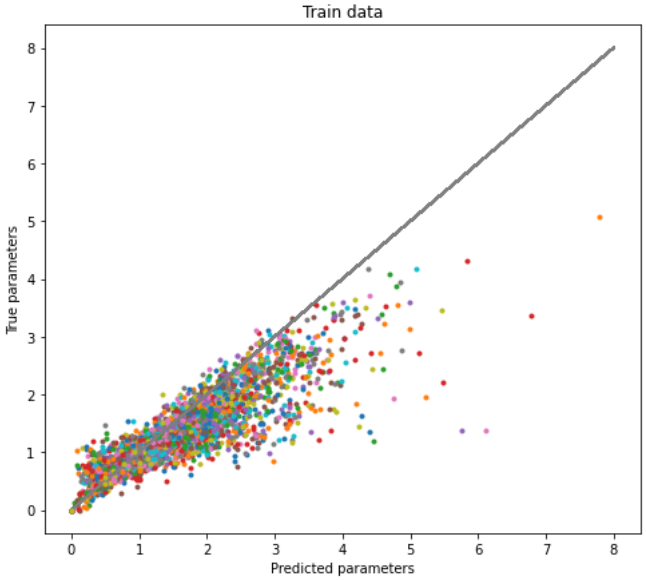

Figure C.8: True data vs predicted data for the train set in case of autorepressor model. RNA+PROTEINS autocorrelations are considered as input. Dropout is added in the Neural Network model.

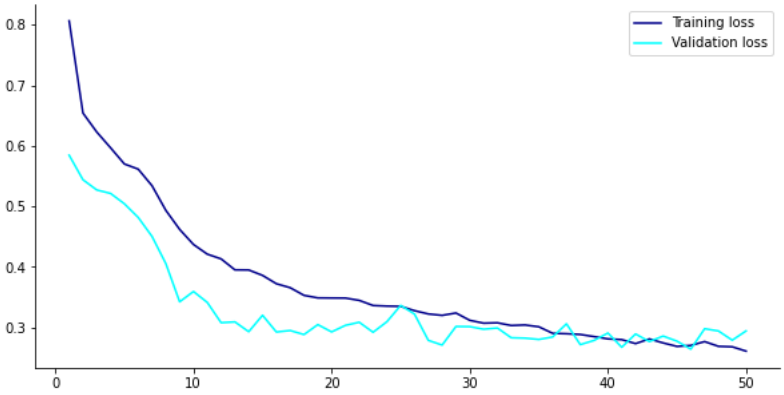

Figure C.9: Learning curve of the ANN refered to only RNAS+PROTEINS autocorrelations as input for the autorepressor model. Dropout is added in the Neural Network model.

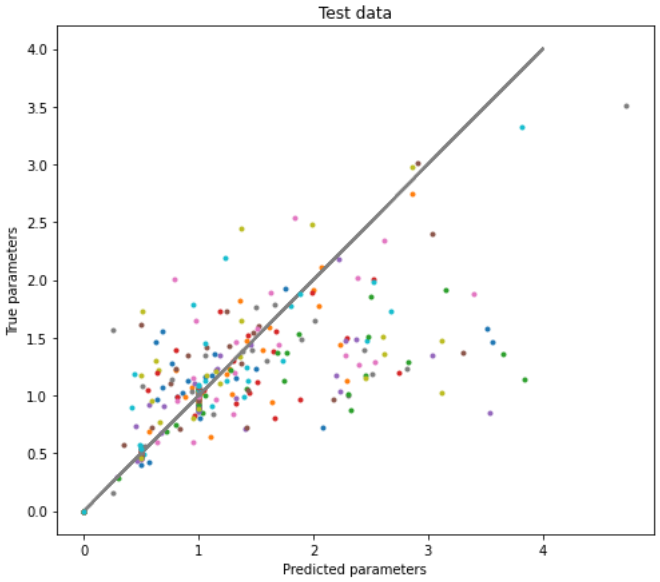

Figure C.10: True data vs predicted data for the test set in case of autorepressor model. RNA autocorrelations are considered as input. Dropout is added in the Neural Network model.

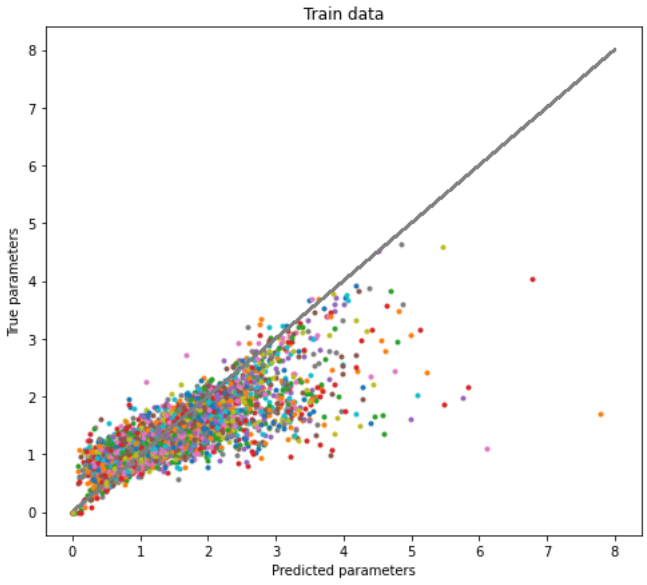

Figure C.11: True data vs predicted data for the train set in case of autorepressor model. RNA autocorrelations are considered as input. Dropout is added in the Neural Network model.

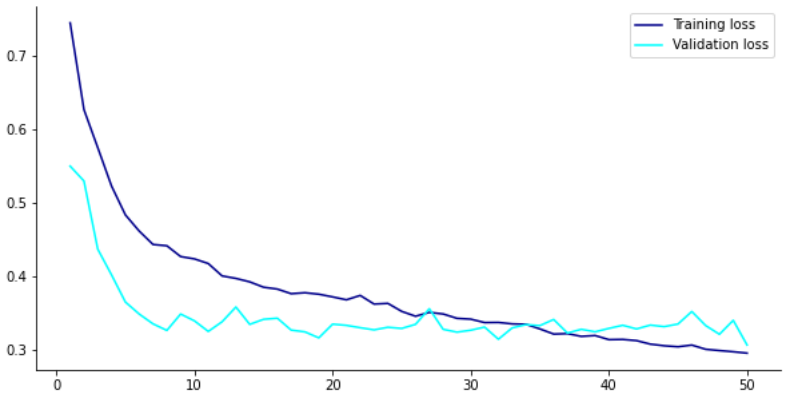

Figure C.12: Learning curve of the ANN refered to only RNA autocorrelations as input for the autorepressor model. Dropout is added in the Neural Network model.

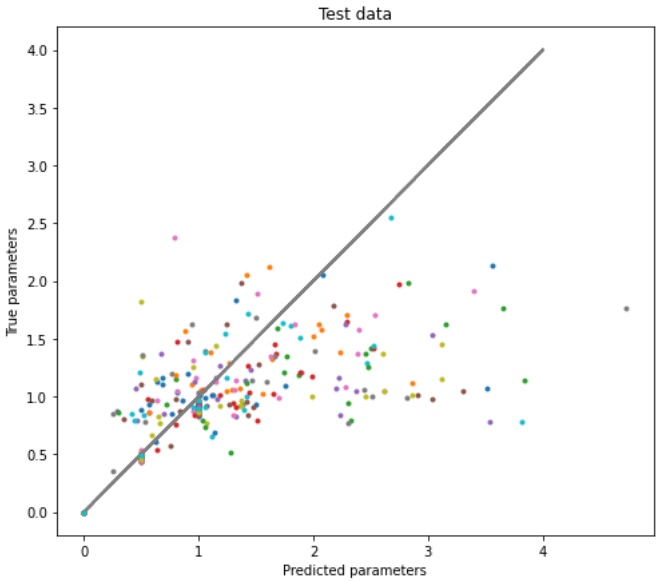

Figure C.13: True data vs predicted data for the test set in case of autorepressor model. PROTEINS autocorrelations are considered as input. Dropout is added in the Neural Network model.

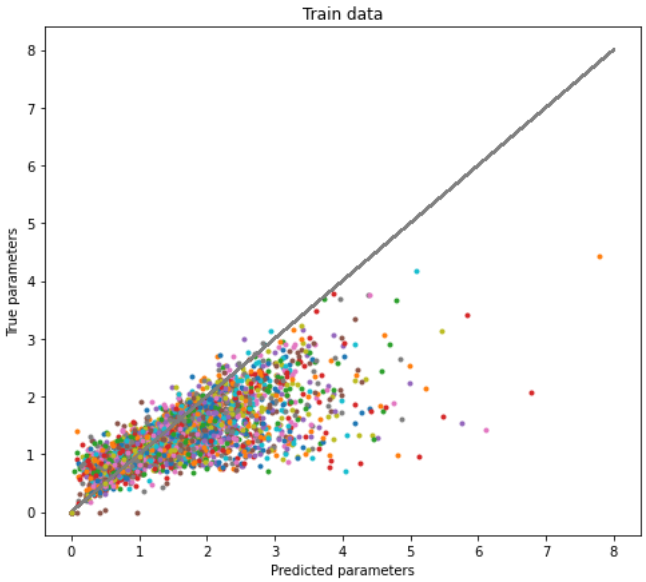

Figure C.14: True data vs predicted data for the train set in case of autorepressor model. PROTEINS autocorrelations are considered as input. Dropout is added in the Neural Network model.

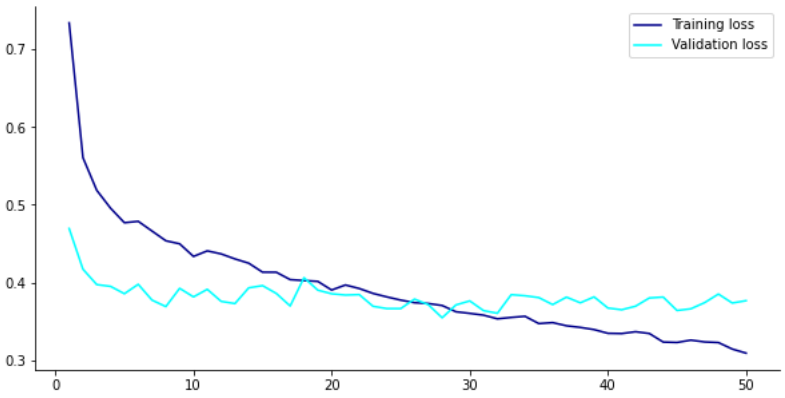

Figure C.15: Learning curve of the ANN refered to only proteins autocorrelations as input for the autorepressor model. Dropout is added in the Neural Network model.

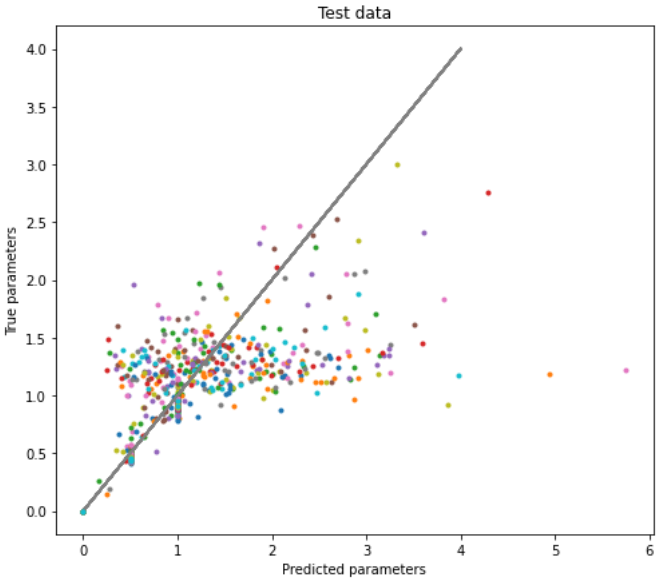

Figure C.16: True data vs predicted data for the test set in case of toggle switch model. RNA autocorrelations are considered as input. Dropout is added in the Neural Network model.

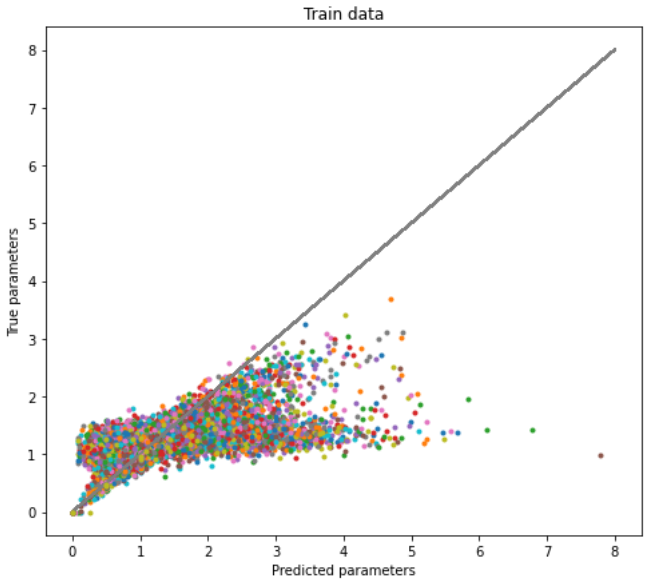

Figure C.17: True data vs predicted data for the train set in case of toggle switch model. RNA autocorrelations are considered as input. Dropout is added in the Neural Network model.

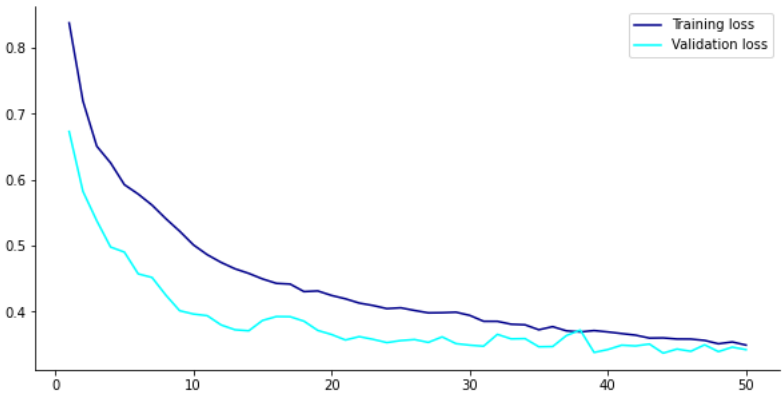

Figure C.18: Learning curve of the ANN refered to only RNA autocorrelations as input for the toggle switch model. Dropout is added in the Neural Network model.

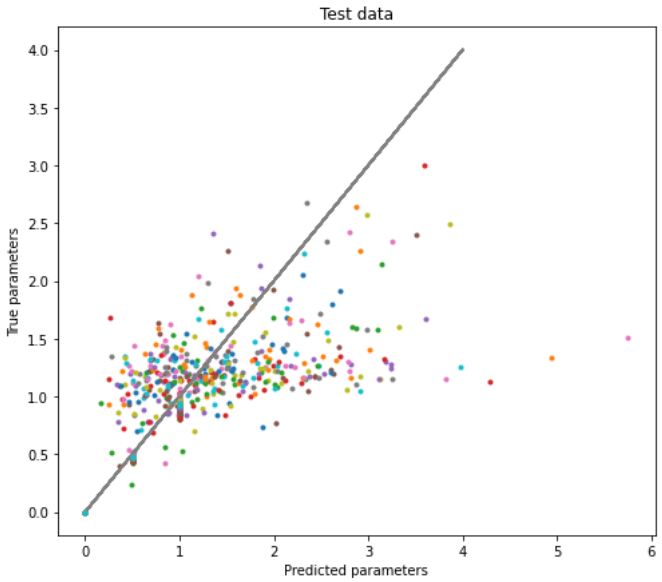

Figure C.19: True data vs predicted data for the test set in case of toggle switch model. PROTEINS autocorrelations are considered as input. Dropout is added in the Neural Network model.

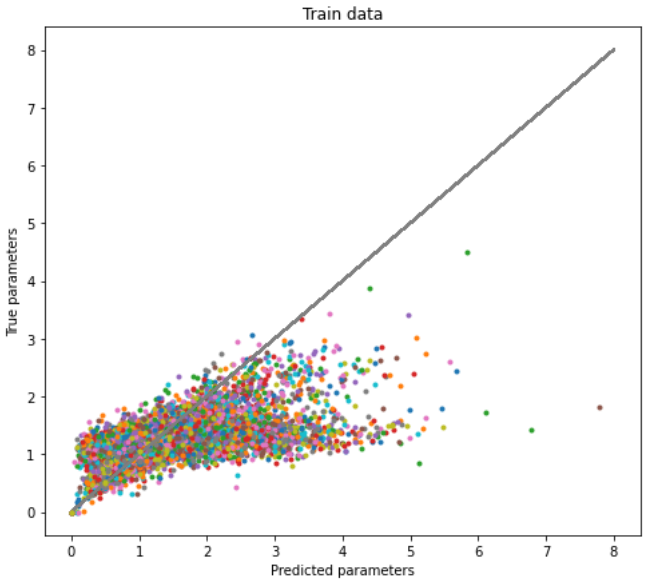

Figure C.20: True data vs predicted data for the train set in case of toggle switch model. PROTEINS autocorrelations are considered as input. Dropout is added in the Neural Network model.

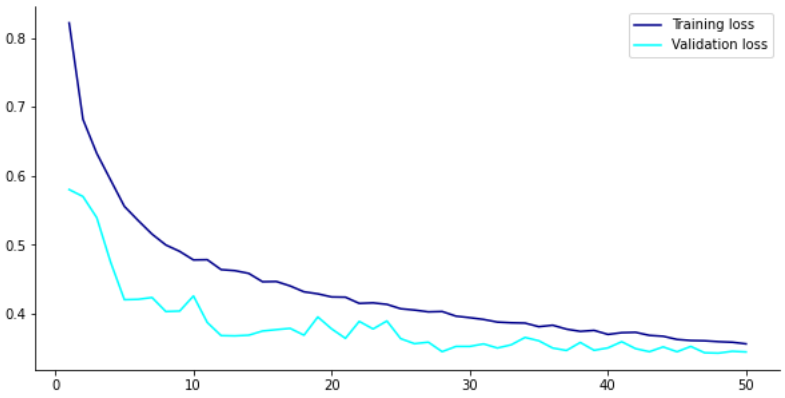

Figure C.21: Learning curve of the ANN refered to only PROTEINS autocorrelations as input for the toggle switch model. Dropout is added in the Neural Network model.

Appendix D

NF-kB activity time series

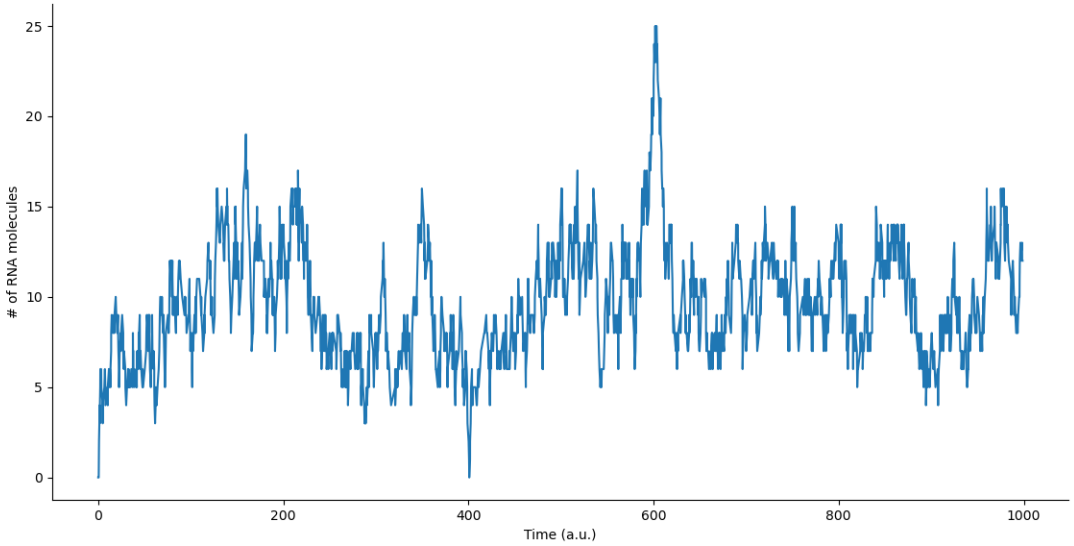

Figure D.1: Number of RNA molecules produced through the NF-κB activation and inhibition simulated by using the Stochastic Simulation Algorithm and considering the model in Fig.9.1 with parameters $ka = 1$,$ki = 0.1$,$k1 = 1$,$k1i = 0.1$,$k4 = 1$,$k5 = 0.1$,$k3 = 0.1$,$k3i = 1$ but considering $k2 = 0$ and $k2i = 0$ starting from a configuration where everything is inactive.

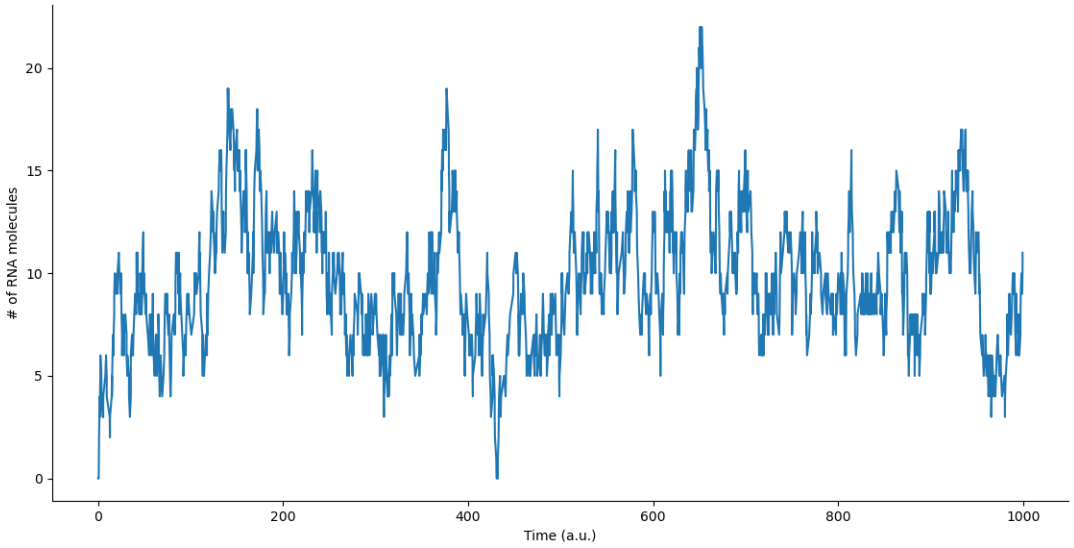

Figure D.2: Number of RNA molecules produced through the NF-κB activation and inhibition simulated by using the Stochastic Simulation Algorithm and considering the model in Fig.9.1 with parameters $ka = 1$,$ki = 0.1$,$k1 = 1$,$k1i = 0.1$,$k4 = 1$,$k5 = 0.1$, but considering $k2 = 0$, $k2i = 0$, $k3 = 0$, $k3i = 0$ starting from a configuration where everything is inactive.

Acknowledgments

I gratefully acknowledge the help and support of my Supervisor, Prof. Enrico Giampieri. His knowledge, experience and advices, together with his availability and encouragements, gave me the possibility to study on this thesis.

I also would like to acknowledge the support and help from all the students in Applied Physics. The year spent for laboratories and lessons was really nice.

Last, but not least, I acknowledge my family for their continuous love and support. I aknowledge also my friends for their encouragements.

After all, if we are what we are and we achieve what we achieve is also due to the environment in which we live and so to the people with which we interact (as explained in Chapter 1.1).

Bibliography

[1] Gastone Castellani, teaching material of *Physical Methods of Biology* course, University of Bologna.

[2] Daniel Remondini, Gastone Castellani, teaching material of *Pattern Recognition* course, University of Bologna.

[3] Claudia Sala, Maximiliano Sioli, teaching material of *Statistical Data Analysis for Applied Physics* course, University of Bologna.

[4] Enrico Giampieri, teaching material of *Software and Computing for Applied Physics* course, University of Bologna.

[5] Giampieri Enrico, *Stochastic models and dynamic meaures for the characterization of bistable circuits in cellular biophysics*, University of Bologna (2012).

[6] Davide Giosué Lippolis, *Stochastic modeling of fluctuations in the NF-κB activity of neoplastic cells*, University of Bologna (2018/2019).

[7] N. G. van Kampen, *Stochastic Processes in Physics and Chemistry*, North Holland, third ed., May 2007.

[8] Yifeng Xia, Shen Shen, and Inder M. Verma, *NF-κB, an active player in human cancers*, Cancer Immunol Res (2014) 2 (9): 823–830.

[9] NF-kB Transcription Factors. URL `https://www.bu.edu/nf-kb/`.

[10] Rel homology domain (RHD), DNA-binding domain. URL `https://www.ebi.ac.uk/interpro/entry/InterPro/IPR011539/`.

[11] NF-kB. URL `https://en.wikipedia.org/wiki/NF-%CE%BAB`.

[12] Tamás Székely Jr., Kevin Burrage, *Stochastic simulation in systems biology*, Computational and Structural Biotechnology Journal, Volume 12, Issues 20–21, November 2014, Pages 14-25.

[13] Yang Cao, Daniel T. Gillespie and Linda Petzold, *Avoiding Negative Populations in Explicit Poisson Tau-Leaping*, Article in The Journal of Chemical Physics, September 2005.

[14] INFLAMMATION'S ROLE IN OBESITY. URL `https://sites.tufts.edu/hkerstjaaalislai/?page_id=541`.

[15] Kateryna Shostak and Alain Chariot, *EGFR and NF-kB: partners in cancer*, Trends in Molecular Medicine June 2015, Vol. 21, No. 6.

[16] NF-κB pathways, Part 1: The canonical pathway of NF-κB activation. URL `https://www.youtube.com/watch?v=7F_HPRfHdDk`.

[17] Genetic Heterogeneity. URL https://www.sciencedirect.com/

[18] epidermal growth factor receptor. URL https://www.cancer.gov

[19] growth factor. URL https://www.cancer.gov

[20] Epidermal growth factor. URL https://en.wikipedia.org

[21] N. G. van Kampen, *Stochastic Processes in Physics and Chemistry*, North Holland, third ed., May 2007.

[22] Autocorrelation Function. URL https://www.sciencedirect.com

[23] G. Cowan, *Statistical Data Analysis*, Oxford University Press Inc., New York, 1998.

[24] Mathisca de Gunst, Michel Mandjes and Birgit Sollie, *Statistical inference for a quasi birth–death model of RNA transcription*, BMC Bioinformatics 23, 105 (2022).

[25] C.W.Gardiner, *Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences*, Printed in Germany, Springer-Verlag Berlin Heidelberg 1983,1985.

[26] Luciana Renata De Oliveira, *Master Equation: Biological Applications and Thermodynamic Description*, University of Bologna (2013/2014).

[27] A guide for using the Wavelet Transform in Machine Learning. URL https://ataspinar.com

[28] Morlet Wavelet. URL https://www.sciencedirect.com

[29] Michael X Cohen, *A better way to define and describe Morlet wavelets for time-frequency analysis*,Radboud University and Radboud University Medical Center; Donders Institute for Neuroscience.

[30] Wavelets: a mathematical microscope. URL https://www.youtube.com

[31] Yoshiteru Sasaki, Kazuhiro Iwai, *Roles of the NF-κB Pathway in B-Lymphocyte Biology*, Curr Top Microbiol Immunol. 2016;393:177-209.

[32] Immunoglobulin Light Chain. URL https://www.sciencedirect.com

[33] Kumar A, Balbach J., *Folding and Stability of Ankyrin Repeats Control Biological Protein Function*. Biomolecules. 2021 Jun 5;11(6):840. doi: 10.3390/biom11060840. PMID: 34198779; PMCID: PMC8229355.

[34] Li J, Mahajan A, Tsai MD., *Ankyrin repeat: a unique motif mediating protein-protein interactions*. Biochemistry. 2006 Dec 26;45(51):15168-78. doi: 10.1021/bi062188q. PMID: 17176038.

[35] hematopoietic stem cell. URL https://www.cancer.gov

[36] Knockout. URL https://www.genome.gov

[37] Zook EC, Kee BL., *Development of innate lymphoid cells*. Nat Immunol. 2016 Jun 21;17(7):775-82. doi: 10.1038/ni.3481. PMID: 27328007.

[38] transinhibition. URL https://en.wiktionary.org

[39] B cell. URL https://www.cancer.gov

[40] Cheong R, Hoffmann A, Levchenko A., *Understanding NF-kappaB signaling via mathematical modeling*. Mol Syst Biol. 2008;4:192.

[41] Ch. Venkateswarlu, Rama Rao Karri, *Optimal State Estimation for Process Monitoring, Fault Diagnosis and Control*, Chapter 5 - Data-driven modeling techniques for state estimation, Elsevier, 2022, Pages 91-111.

[42] Genetic Heterogeneity. https://en.wikipedia.org

[43] M. Manceau, V.S. Domingues, C.R. Linnen, E.B. Rosenblum, H.E. Hoekstra, *Convergence in pigmentation at multiple levels: mutations, genes and function*. Phil Trans R Soc B, 365 (2010), pp. 2439-2450.

[44] D. Stockholm, R. Benchaouir, J. Picot, P. Rameau, T.M.A. Neildez, G. Landini, et al., *The origin of phenotypic heterogeneity in a clonal cell population in vitro*. PLoS One, 2 (2007), p. e394.

[45] Close J.L. Spudich, D.E. Koshland Jr., *Non-genetic individuality: chance in the single cell*. Nature, 262 (1976), pp. 467-471.

[46] Munsky B, Neuert G, van Oudenaarden A.,*Using gene expression noise to understand gene regulation*. Science. 2012 Apr 13;336(6078):183-7.

[47] M. Viney, S.E. Reece, *Adaptive noise*. Proc R Soc B, 280 (2013), p. 20131104.

[48] A. Bar-Even, J. Paulsson, N. Maheshri, M. Carmi, E. O'Shea, Y. Pilpel, et al., *Noise in protein expression scales with natural protein abundance*. Nat Genet, 38 (2006), pp. 636-643.

[49] Matt Scott, *Tutorial: Genetic circuits and noise*. Quantitative Approaches to Gene Regulatory Systems. Summer School, July 2006. University of California, San Diego.

[50] Housekeeping gene https://en.wikipedia.org

[51] Zenklusen D, Larson DR, Singer RH. *Single-RNA counting reveals alternative modes of gene expression in yeast*. Nat Struct Mol Biol. 2008 Dec;15(12):1263-71.

[52] Gene regulation. https://www.genome.gov

[53] RFX1 Gene - Regulatory Factor X1 https://www.genecards.org

[54] Issac, J., Raveendran, P.S. Das, A.V., *RFX1: a promising therapeutic arsenal against cancer*. Cancer Cell Int 21, 253 (2021).

[55] Lubelsky Y, Reuven N, Shaul Y.,*Autorepression of rfx1 gene expression: functional conservation from yeast to humans in response to DNA replication arrest*. Mol Cell Biol. 2005 Dec;25(23):10665-73.

[56] Giampieri E, Remondini D, de Oliveira L, Castellani G, Lió P., *Stochastic analysis of a miRNA-protein toggle switch*. Mol Biosyst. 2011 Oct;7(10):2796-803.

[57] Random Process (or Stochastic Process) https://uotechnology.edu.iq

[58] Zhang Q, Lenardo MJ, Baltimore D., *30 Years of NF-?B: A Blossoming of Relevance to Human Pathobiology*. Cell. 2017 Jan 12;168(1-2):37-57.

[59] Oncogene https://en.wikipedia.org

[60] Manuel D. Rossetti, *Simulation Modeling and Arena*. https://rossetti.github.io/RossettiArenaBook/

[61] Runge Kutta methods https://en.wikipedia.org/wiki/Runge-Kutta_methods

[62] Harmonic analysis and the Fourier Transform https://terpconnect.umd.edu

[63] Ralston, A. Shaw, K. (2008) *Environment controls gene expression: Sex determination and the onset of genetic disorders*. Nature Education 1(1):203.

[64] Ali MZ, Brewster RC (2022) *Controlling gene expression timing through gene regulatory architecture*. PLOS Computational Biology 18(1): e1009745.

[65] Wang ZR, Wang L, Wan CM, Cornelissen G, Anand I, Halberg F. *Circadian rhythm of gene expression of myocardial contractile protein, left ventricular pressure and contractility*. Space Med Med Eng (Beijing). 1999 Dec;12(6):391-6. PMID: 12432879.

[66] Synchrosqueezing in Python https://github.com/OverLordGoldDragon/ssqueezepy

[67] Synchrosqueezing Wavelet Transform explanation? (How's it work, exactly?) https://dsp.stackexchange.com

[68] Wavelet "center frequency" explanation? Relation to CWT scales? https://dsp.stackexchange.com

[69] How is wavelet center frequency computed? https://dsp.stackexchange.com

[70] tumor necrosis factor https://www.cancer.gov

[71] A process in which a phosphate group is added to a molecule, such as a sugar or a protein. https://www.cancer.gov

[72] Nestler EJ, Greengard P. *Protein Phosphorylation is of Fundamental Importance in Biological Regulation*. In: Siegel GJ, Agranoff BW, Albers RW, et al., editors. Basic Neurochemistry: Molecular, Cellular and Medical Aspects. 6th edition. Philadelphia: Lippincott-Raven; 1999.

[73] Tomasz Lipniacki, Pawel Paszek, Allan R. Brasier, Bruce Luxon, Marek Kimmel, *Mathematical model of NF-kB regulatory module*, Journal of Theoretical Biology, Volume 228, Issue 2, 2004, Pages 195-215, ISSN 0022-5193.

[74] Hill equation (biochemistry) `https://en.wikipedia.org/wiki/Hill_equation_(biochemistry)`

[75] Synaptic Strength https://www.sciencedirect.com

[76] H. H. Sultan, N. M. Salem and W. Al-Atabany, *Multi-Classification of Brain Tumor Images Using Deep Neural Network*, in IEEE Access, vol. 7, pp. 69215-69225, 2019.

[77] (2011). Mean Absolute Error. In: Sammut, C., Webb, G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA.

[78] scipy.stats.gamma https://docs.scipy.org