MODELING AND EVOLUTIONARY ANALYSIS OF GENE REGULATORY NETWORKS

by

Seher Uğurcuklu

B.S. in Physics
Boğaziçi University, 2012

Submitted to the Institute of Biomedical Engineering
in partial fulfillment of the requirements
for the degree of
Master of Science
in
Biomedical Science

Boğaziçi University
2014

MODELING AND EVOLUTIONARY ANALYSIS OF GENE REGULATORY NETWORKS

APPROVED BY:

Assoc. Prof. Dr. Albert Güveniş (Thesis Advisor)	....
Assist. Prof. Dr. Cem Özen (Thesis Co-advisor)	....
Assist. Prof. Dr. Bora Garipcan	....
Assoc. Prof. Dr. Burak Güçlü	....
Assoc. Prof. Dr. Mehmet Burçin Ünlü	....

DATE OF APPROVAL: August 1, 2014

ACKNOWLEDGMENTS

First, I would like to express my sincere gratitude to my co-advisor Dr. Cem Özen for the patience, tolerance and helpful criticisms he provided throughout the thesis and his crucial role in my ongoing interest in the multi-disciplinary field.

Also, I would like to thank my advisor Dr. Albert Güveniş for his interest and guidance. I am grateful to Dr. Burçin Ünlü for having guided me into biomedical sciences and his encouraging conducts. For their helpful instructions in developing my background knowledge and computational skills, I am thankful to Dr. Taylan Akdoğan and Dr. Arzucan Özgür.

Many thanks to my friends at the institute for the good time spent together. My heartfelt thanks to all of my dear friends, Emre Çelebi, Eda Aydın, Cem Eröncel, Roya Nouri Rad for their company and kindliness.

My special gratitude goes to my dear friend Fadime Şener whose love, inspiration and support mean so much to me.

Finally, I would like to thank my dear family for their care, support and love throughout my life.

ABSTRACT

MODELING AND EVOLUTIONARY ANALYSIS OF GENE REGULATORY NETWORKS

In Systems Biology and more recently emerging field of Synthetic Biology, mathematical modeling has become an indispensable component of research. As complementary to the experimental studies, computer simulations are used to accelerate the hypothesis generation-validation cycle of research in biological systems.

This thesis is mainly concerned with modeling and inference of gene regulatory dynamics on the basis of gene expression patterns. At first, we make a statistical analysis over randomly generated genetic networks, based on their oscillatory dynamics. Then, in our model problem, we aim to design a family of genetic networks that exhibit stable periodic oscillations with a prescribed period. Later, we investigate the temporal behaviour of a system utilizing a computer simulation. We design such circuits on the basis of *in silico* evolution of the corresponding network model.

The approach starts with a randomized gene network. Then, structural rewiring mutations are applied to the networks. In this process, evolving networks are selected depending on their closer approach to the targeted dynamics, after a mutation. By using this method, networks with required oscillation periods are constructed by changing the architecture of regulatory connections between the genes.

In addition, we choose a small genetic network that exhibits chaotic dynamics, and look at the change of its dynamics against a system parameter. Such an approach is useful in deriving the characteristics of these systems under specific variations.

Keywords: Dynamical Systems, Gene Regulatory Networks, Evolutionary Algorithm, Mathematical Modeling, Inference, Optimization.

ÖZET

GENETİK DÜZENLEYİCİ AĞLARIN MATEMATiKSEL MODELLENMESİ VE EVRİMSEL ANALİZİ

Matematiksel modelleme yöntemleri, Sistem Biyolojisi ve Sentetik Biyoloji alanındaki araştırmaların vazgeçilmez bir unsuru haline gelmiştir. Biyolojik sistemleri inceleyen bilgisayar simülasyonu çalışmaları, deneysel çalışmalara tamamlayıcı bir yöntem olarak hipotezin geliştirilmesi-doğrulanması döngüsünde kullanılmaktadır.

Bu çalışma, gen ekspresyonu modellerini baz alan genetik düzenleyici dinamiklerin modellenmesi ve analizi hakkındadır. Çalışmamıza rastgele oluşturulmuş gen ağlarının osilasyon dinamiğine bağlı bir istatistiksel analizle başlıyoruz. Sonrasında stabil periyodik osilasyonlar yapan bir grup gen sistemi tasarlayarak, bilgisayar simülasyonu aracılığıyla bu sistemlerin davranışındaki zamana bağlı gelişimleri inceliyoruz.

Araştırmamız, genlerin regülasyonunun aktivasyon ya da baskılanma etkileşimiyle sağlandığı, rastgele olarak oluşturulmuş bir gen sistemi ile başlatılıyor. Sonrasında, genler arası etkileşimsel mutasyonlar uygulanıyor. Mutasyonlar sonrası dönüşme uğrayan sistemler, bir evrimsel algoritma aracılığıyla hedeflenen dinamiklere yakınsamalarına bakılarak bir seçime tabii tutuluyor. Bu yöntemle, genler arası düzenleyici yapılar dönüştürülerek, hedeflenen osilasyon periyoduna sahip sistemler inşa edilmiş oluyor.

Ek olarak, kaotik dinamiğe sahip bir gen modeli ele alarak, modeldeki bir parametre varyasyonuna karşılık sistemdeki dinamik değişiklikleri gözlemliyoruz. Bu yöntemle, spesifik varyasyonlar altındaki benzer sistemlerin karakteristiğinin nasıl belirlenebileceğini gösteriyoruz.

Anahtar Sözcükler: Dinamik Sistemler, Genetik Düzenleyici Ağlar, Evrimsel Algoritma, Matematiksel Modelleme, Çıkarsama, Optimizasyon.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

DNA	deoxyribonucleic acid
RNA	ribonucleic acid
mRNA	messenger RNA
RNAP	RNA polymerase
TF	transcription factor
GRN	gene regulatory network
M-H	Metropolis-Hastings

1. INTRODUCTION

All living organisms exhibit a complex hierarchical organisation of small building blocks. The whole information required for the functioning of a cell is encoded in the DNA sequence that is passed on from one cell to another in inheritance by *gene*s. Expression of the genes leads to formation of *proteins*, then a combination of these proteins defines the specific functionality of the cell. Malfunctioning at any stage of these building blocks may cause a living organism to stop performing normally leading to a diseased state, and the most causes of diseases can be mapped to the abnormal activity of some genes in the cells.

It has been discovered in the early sixties that, some proteins can regulate (by activating/inhibiting themselves or other genes) the expression of genes in a living organism. ([2],[3]) Furthermore, the researchers realized that these regulations of gene expression occur through complex networks of regulatory interactions in a nonlinear way between genes, mRNAs, proteins and small molecules. Understanding the complex networks of gene regulatory interactions is still in progress. Mathematical modeling and computer science techniques are the essential tools to understand these complicated interactions effectively.

The Human Genome project [4], one of the primary aims of which was to identify all protein coding genes, has estimated and identified approximately $20,000-25,000$ protein coding genes in humans. With the identification of all protein coding genes, various high-throughput technologies (such as DNA microarrays [5], protein arrays [6]) have emerged since the completion of the project. These technologies can measure the expression (or activity) of all genes in a genome simultaneously. By measuring and comparing the expression of genes in an unhealthy vs. a healthy cell, it is now possible to identify genes responsible for various diseases at the entire genome level.

On the other hand, these high-throughput technologies have lead to the generation of enormous amounts of experimental data. However, this is a slow and exhaustive process that fails to adequately approach the true complexities of living phenomena and is of limited relevance to biological systems as a whole. With such an amount of new biological data from research labs, a new branch of research is emerged for the interpretation of data in the most efficient manner ([7], [8], [9]). Many advanced statistical and computational tools have been developed to help biologists identify novel targets from their experimental data.

Studying the behavior of genes with respect to other genes (or proteins) that are known to play a role in a specific disease, is critical to manipulate these genes using drug molecules. So, the focus of computational and experimental tools has shifted from just measuring the expression of genes, to construction of gene-gene, gene-protein and protein-protein interaction networks ([10], [8], [9]). New computational methods are being developed to model and analyse these interaction networks.

Interaction networks can represent the dynamic behaviour in terms of the flow of signaling from a biological entity to another and are named such as *signaling pathways*, *gene regulatory networks* (GRNs). These GRNs can be seen as dynamical systems of sets of genes and proteins where the gene (or protein) expression is a function of the expression of other genes (and proteins) with which it can directly interact [11] . By modeling the dynamical system represented by GRNs, it is possible to make an *in silico* simulation of the evolution of a gene (and protein) expression over time. Also, one can study how the system behaves when it is slightly changed to reflect gene mutations inside a cell.

For modeling the behavior of a cell, one could either gather the crucial information on the precise biochemical processes or choose to model the flow of information in gene interaction networks. In this dissertation, we choose the latter approach to identify qualitative behavior of the biological system under study.

Qualitative modeling of GRNs has been a research interest to theoretical biologists for at least last four decades ([12], [13], [14], [15]). Most of the initial focus of qualitative modeling was on studying theoretical properties of the dynamical nature of GRNs rather than developing computational methods for modeling large GRNs, as for this study. However in the last decade, a need for efficient computational tools for qualitative modeling of GRNs has been arised so as to understand the experimental data in the context of the dynamical behavior of a cell ([16], [17]).

1.1 Systems Biology and Synthetic Biology

Cells, tissues, organs, organisms and ecological webs are systems of components with specific interactions, so a system-level understanding should be the primary achievement of biology. The purpose of *Systems Biology* is to describe these interactions in precise quantitative mathematical terms that allow making predictions about how cells may respond to external and internal perturbations [18]. So, it integrates quantitative experimental results, then builds predictive models.

Systems biology descriptions rely on the modeling and the simulation of so-called networks, loosely defined as ensembles of biological objects (such as genes, RNAs, proteins, etc.) interacting with each other. One advantage of this approach is that it considerably simplifies the problem of conecting networks structure and kinetics to biological behaviour, while being complex enough to have access to a wide range of realistic dynamics.

Systems biology can proceed in two directions. A “bottom-up” approach, examines the mechanisms through which functional properties arise in the interactions of known components. Thus, a mathematical formalism describes the changes in concentration of each gene transcript and protein as a function of their regulatory interactions. Such a network can then be used to probe the behaviour of the biological process using computer simulations and mathematical analysis, to generate novel hypotheses to be then tested *in vivo*. The “top-down” approach, on the contrary, is called “reverse engineering" and, typically, identifies molecular interaction networks based on correlated molecular behavior that observed in gene-gene interactions [19].

In this way, systems biology integrates experimental and modeling approaches to explain the structural and functional organization of complex biological systems as networks of dynamic interactions [20]. Over the last two decades, mathematical and computational techniques have been widely applied in biology aiming to understand how cells behave and differentiate as the result of tens of thousands of molecular interactions. So, computational systems biology, through theoretical exploration and modeling, provides a powerful foundation from which to address critical scientific questions [8].

Furthermore, advances in molecular manipulation techniques and accumulation of genetic information, are progressively opening new possibilities for gene therapy and biomedical engineering. By combining naturally occurring genetic components in novel ways, it has become possible to artificially engineer genetic networks with sophisticated functional capabilities. The desired characteristics of such networks can be rationally designed and tested through predictive modeling. This emerging discipline of *Synthetic Biology* can be defined as the engineering of biology.

Similar to electrical networks, genetic networks possess "input" and "output" functionality such that they are capable of responding in highly specified mechanisms. The effective control of gene expression leads to the development of regulatory circuits with a broader range of functional behaviors. Functions like threshold-based switching, and oscillatory or sequential expression, are promising approaches to construct autonomous synthetic regulatory systems.

Synthetic biology relies on mathematical modeling, informatics and control theory; and shares tools from genetic engineering, bioengineering, systems biology, and many other engineering disciplines. Synthetic biology studies does not only investigate the effects of genetic and pathway modification or the cellular responses on genetic variation/environmental perturbation, but also design and build biological systems with novel cellular functions, combining *in silico* and *in vivo* experimental approaches [18].

Genetic networks have been studied theoretically and experimentally for many years, usually by focusing on analysis of naturally occurring circuits. Two pioneering papers have reported the *de novo* design of a functioning biological circuit using well-characterized genetic elements. The toggle switch [21] forms a synthetic, addressable cellular memory unit and has implications for biotechnology, biocomputing and gene therapy. Repressilator network [22], on the other hand, was implemented in *Escherichia coli* and designed to exhibit a stable oscillation which is reported via the expression of green fluorescent protein, and acts like an electrical oscillator system with fixed time periods. Both circuits were designed with the aid of mathematical models, and their results show that it is possible to design and construct an artificial genetic network with new functional properties.

Such synthetic circuits are used *in vivo* models to explore the relation between the structure and function of a genetic circuit [23]. Within a short period of time, these rational model-guided approaches have turned out to be very fruitful for engineering standardized genetic modules for biotechnological applications from a library of characterized promoters [24].

A recent survey [25] resumes two periods of synthetic biology by pointing out both *in silico* and *in vivo* approaches. Accordingly, in the first wave of synthetic biology, basic elements are combined to form small modules with specified behaviours. Such modules include switches, cascades, pulse generators, time-delayed circuits, oscillators, spatial patterning and logic formulas which can be used to regulate gene expression, metabolism and cell-cell communication. Then, on the second wave of synthetic biology, these basic parts and modules are integrated to create systems-level circuitry.

The utility of a model in both Systems and Synthetic biology lies in its ability to formalise the knowledge about the biological process at hand, to identify inconsistencies between hypotheses and observations, and to predict the behaviour of the biological process in yet untested cases. *In silico* studies of the behaviour of a biological system is nowadays often used to complement *in vivo* experimental observations and accelerate the hypothesis generation-validation cycle of research [26].

1.2 Gene Regulation

The cell is the basic structural and functional unit of all known living organisms. The nature of a cell and its functions are mainly characterized by its DNA. DNA is a nucleic acid that contains the genetic instructions used in the development and functioning of the organisms. The whole hereditary information that is encoded either in the DNA or, for some virus, in RNA.

DNA has two strands, the coding strand and the noncoding strand. The strand of DNA that does not carry the information necessary to make proteins is called non-coding DNA. Only the coding strand contains the information for making proteins. The DNA segment carrying the genetic information to produce a polypeptide chain is called genes. Thus, the genome includes both the genes and the non-coding sequences of the DNA/RNA.

The so-called *central dogma* of molecular biology comprises the three major processes: *replication*, *transcription*, and *translation*. *Replication* is the copying of parental DNA to form daughter DNA molecules with identical nucleotide sequences. *Transcription* is the process by which the genetic information stored in DNA sequence is copied precisely into messenger RNA. *Translation* is the genetic information encoded in messenger RNA being translated into a protein polypeptide. These processes are illustrated in Figure 1.1.

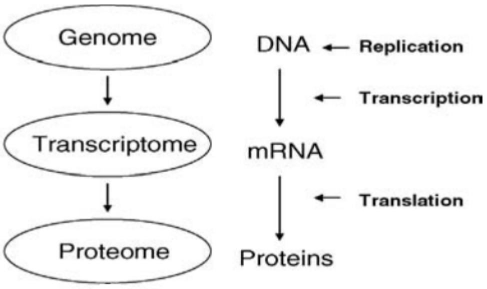


Figure 1.1: The classical central dogma definition at whole genome level [27].

Proteins or protein complexes are involved in almost all biochemical processes of a living organism. For example, they are essential for enzymatic catalyzation of biochemical reactions, cell signalling, identifying and neutralizing extraneous objects in cells as an immune response, regulating transcriptional processes of genes, maintaining cell shape in the cytoskeleton, and many other functions.

DNA is composed of a sequence of the four nucleotides Adenine, Cytosine, Guanine, and Thymine. The prescription for building proteins is in the specific regions of DNA, are defined as genes. This information is transcribed and translated into polypeptides which in turn are transformed into functional proteins. In eukaryotic cells, these processes of transcription and translation involves several steps that are illustrated in Figure 1.2.

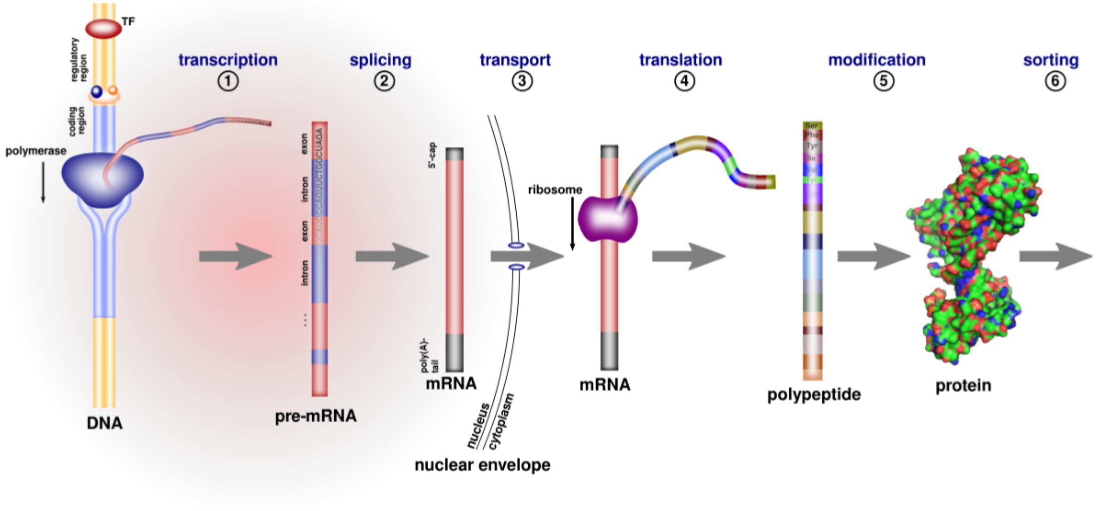


Figure 1.2: Illustration of gene expression in eukaryotic cells [28].

1.2.1 Transcriptional Regulation

Gene expression starts with transcription (Figure 1.2), that is the synthesis of a messenger RNA (mRNA) from a DNA template that have the coding and regulatory regions. Coding sequences of DNA are transcribed into a complementary mRNA by the RNA polymerase. A collection of proteins called transcription factors (TFs) regulates the initiation of transcription. They promote or prevent the binding of the enzyme RNA polymerase by binding to the corresponding regulatory region of the gene. When the transcription is initiated, an mRNA copy can be synthesized.

The mRNA is then translated into a protein, the gene product. The rate of transcription, i.e. the number of mRNA molecules produced per unit time, is controlled by the promoter, a regulatory region of DNA. RNA polymerase binds a specific binding site (DNA sequence) at the promoter, thereby leading to the assembly of a multimolecular transcription system.(Figure 1.3)

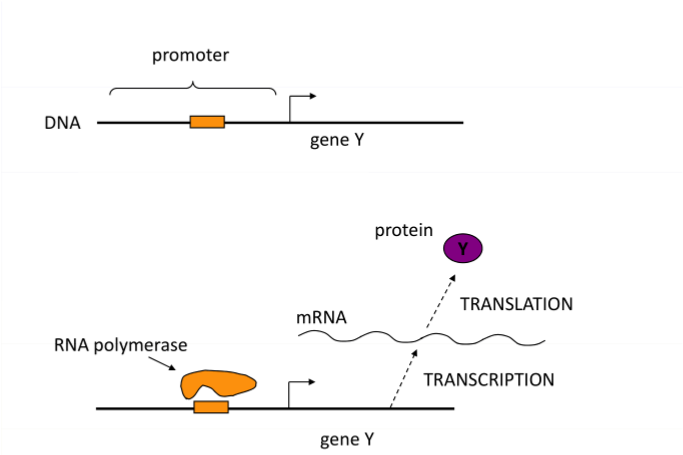


Figure 1.3: Each gene is preceded by a regulatory DNA region called promoter. The promoter contains a specific site that can bind RNA polymerase (RNAp). [18]

After accomplished splicing, the mature mRNA is transported through pores in the nuclear envelope to the cytoplasm (Figure 1.2), where it is translated into the final protein during the next steps.

1.2.2 Feedback Control by Transcription Factors

Genes are regulated by a complex combinatorial interplay of TFs. These factors are gene products and their production is controlled by further TFs. These connections between transcription factors and genes establish an detailed GRN (Figure 1.4). The architecture of this network determines the temporal order of specification events in organisms.

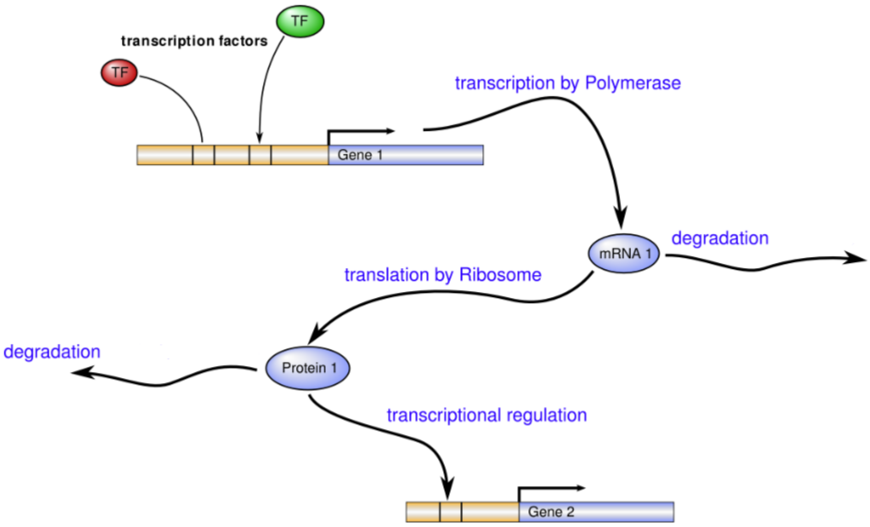


Figure 1.4: Simplified model of transcriptional regulation. Genes are transcribed to mRNAs and then translated into proteins catalyzed by certain enzymes. Some proteins serve as regulators for the transcription of genes. Then, these mRNAs and proteins are degraded. [28]

1.3 Modeling of Gene Regulatory Networks

Genetic oscillatory networks are found in many biological pathways, including the circadian rhythm, cell cycle regulation, apoptosis, metabolism, and morphogenesis. Such networks involve hundreds of reactions and thus are extremely difficult to characterize biologically and mathematically. This highlights the importance of methods to simplify the analysis of these networks.

Interaction networks can represent the dynamic behaviour in terms of the flow of signaling from a biological entity to another and are referred by various names such as signaling pathways, gene regulatory networks or genetic regulatory networks. These GRNs can be seen as dynamical systems of sets of genes and proteins where the gene (or protein) expression is a function of the expression of other genes (and proteins) with which it can directly interact [13].

A GRN can be considered at varying degrees of complexity and is usually described through a collection consisting of genes with regulatory components, nucleic acids, proteins and signaling molecules. In the transcriptional regulatory networks, usually a number of proteins act as regulators (i.e. activators or repressors) on the activity of certain genes by altering their expression rates and thereby changing the production rates of their protein products.

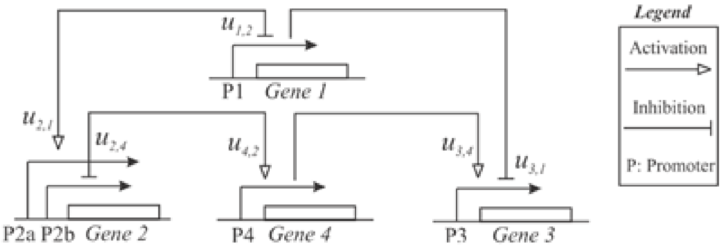


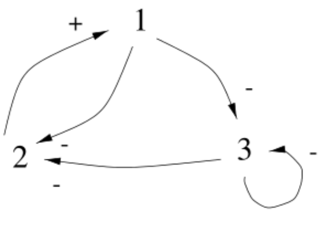
Figure 1.5: An example of a GRN that consists of 4 genes [18].

The graphical representation of interactions between different biological entities is convenient to get directly a general idea and to gain a better global and local understanding of the whole intricate system. How a collection of regulatory proteins associates with genes across a genome can be described as a network in which the nodes are genes and the edges are regulations among them. In Figure 1.5, directed edges with an arrow end represents activation, whereas a dash end represents inhibition.

1.3.1 Mathematical Representation of GRNs

The most straightforward way to model a GRN is to regard it as a *graph*. Formally, a graph is defined as a tuple (V, E), with V indicating a set of *vertices*, and $E \subseteq V \times V$ indicating a set of *edges* as $G = (V, E)$ [29]. The edges represent the relation between vertices and may be directed or undirected. A *directed edge* is a pair $(i, j) \epsilon E$ of vertices, where i denotes the head, and j denotes the tail of the edge. (i, j) is an undirected edge if the order of the vertices is of no importance.

The vertices of a graph correspond to genes or other elements of interest in the cell, while the edges denote interactions among the genes. In the case of directed graphs, edges point from regulating genes to regulated genes, for example, from genes encoding TFs to the targets of the TFs. The graph representation of a GRN can be generalized in several ways. For instance, the vertices and edges could be labeled, by adding information about genes and their interactions. Defining a directed edge as (i, j, s), with s equal to $+$ or $-$, allows one to indicate whether i is activated or inhibited by j, respectively.



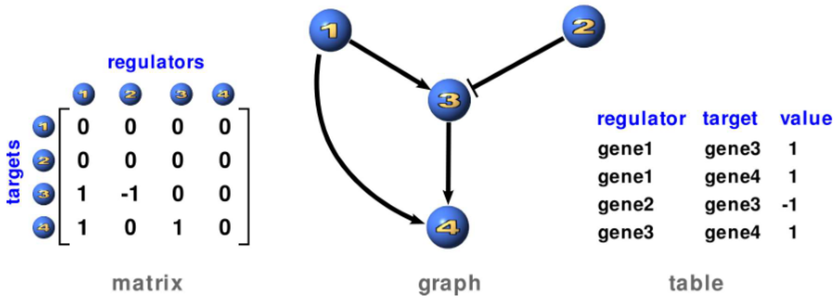
(a)

$$V = \{1, 2, 3\}$$
$$E = \{\langle 2, 1, -\rangle, \langle 3, 1, -\rangle, \langle 1, 2, +\rangle, \langle 2, 3, -\rangle, \langle 3, 3, -\rangle\}$$

(b)

Figure 1.6: (a) Directed graph representing a GRN and (b) its definition.

Many of the pictures of biological networks found in the literature can be mapped to some sort of graph representation. An example of a simple directed graph model is shown in Figure 1.7. This graph is described by a matrix and a list of regulations. Positive and negative effects of regulators on targets are distinguished in the graph by edges with arrows and bars, or in the matrix and table by *1* and *-1*, respectively. Components, i.e., genes, mRNAs, and proteins associated to the same node have the same index $i \, \epsilon \{1, ..., n\}$.



$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 3 & 1 & -1 & 0 & 0 \\ 4 & 1 & 0 & 1 & 0 \end{array}$$

regulator	target	value
gene1	gene3	1
gene1	gene4	1
gene2	gene3	-1
gene3	gene4	1

Figure 1.7: Mathematical representations of a gene regulatory network [28].

Over the last several decades, mathematical modeling of gene circuits have become a mature discipline with a vast literature that includes a number of generally used approaches such as directed graphs, Bayesian networks, Boolean networks, ordinary and partial differential equations, stochastic equations, and rule-based formalisms. In this study, we will be using the ordinary differential equations (ODEs) approach, which is possibly the most often used formalism for the modeling of gene networks.

1.3.2 Simplifying Approximations in Biological Systems Modeling

A mathematical model is a formalization of the biological knowledge about a certain system, where each component of the system is described by an equation, which represents its behaviour as a function of its regulators.

Ideally, all information relevant to a system (concentrations, rates of events, etc.) would be known to make a maximally accurate *in silico* replica of the system. Unfortunately, even for the best-examined systems, the chosen framework fails to meet the standards of describing the variety of elementary processes, even qualitatively.

So, even at the expense of accuracy, assumptions are necessary. It can be beneficial to exclude some known data to accommodate available computational power and to facilitate the analysis as well. Generally, two main approaches (qualitative and quantitative) are utilized to overcome such challenges in modeling (Figure 1.8).

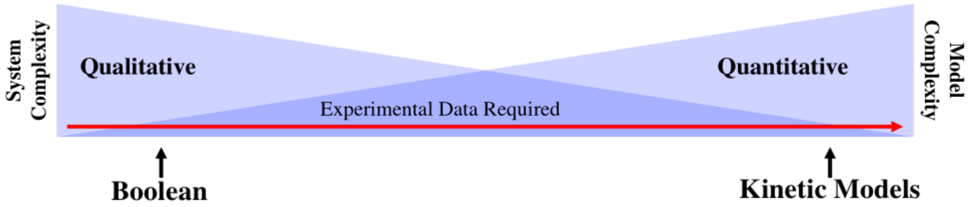


Figure 1.8: Different modeling techniques for simulating GRNs.

Qualitative Models

The simplicity of qualitative models is useful to gain insight into general phenomena of complex systems. So, in qualitative modeling, we need to retrieve from biological data at least the information required for the formulation of logical statements describing, for instance, causal relationships between events involving model components.

In qualitative modeling, kinetic processes are simulated by tracking over discrete time states of the system. The weak specification of such models allows simulations to explore the space of possible behaviours. Moreover, it provides high-level predictions applicable to a whole family of systems. For example, Boolean models can be applied to capture the essentials in a conceptual model including chaotic behavior. A revealing example coming from biology is the supposition of Rene Thomas that multi-stationarity (the existence of multiple steady-states) requires the presence of a positive feedback loop in the network [30].

Quantitative Models

Compared with qualitative models, quantitative ones offer greater detail in mimicking reality. Moreover, rich qualitative insights on the system are possible using theoretical tools such as bifurcation and stability analysis, which, for example, indicate the precise boundaries of parameter ranges to which steady states or sustained oscillations correspond, or reveal the stability of the solutions before actually solving the dynamical equations representing the system. Quantitative models can be either deterministic or stochastic.

Deterministic formalisms are commonly used to describe the average behaviour of a population of cells [31]. They have been shown to be viable for the analysis of synthetic networks in a number of works (e.g. [22], [21], [32], [33], [34]). The reaction mechanism is described by applying the law of mass action: the rate of any given elementary reaction is proportional to the product of the concentrations of the species reacting in the elementary process (reactants) [18].

A better comprehension of the relation between the structure and functioning of a regulatory system calls for the use of dynamical models. ODEs [35] approach is probably the most-widespread formalism for continuous modeling of the dynamical behavior of cellular interaction networks. When ODEs are used, the cellular concentration of proteins, mRNAs and other molecules are represented by continuous time variables. The functional descriptions of transcriptional interactions are usually non-linear Hill function or Michealis-Menten, where the first one differs by considering the cooperativity of a protein on the gene of interest [18].

In this formalism, the concentrations of gene products (mRNAs or proteins) are represented by continuous, time-dependent regulatory variables, $x(t)$, $t \epsilon T$ where T is a closed time interval. The variables take their values from the set of nonnegative real numbers x.

The regulatory interactions between genes are modeled by a system of ODEs having the following general form:

$$\frac{dx_i}{dt} = f_i(x), \qquad i \, \epsilon \{1, ..., n\} \tag{1.1}$$

where $x(t) = \{x_1(t), ..., x_n(t)\}$ is the vector whose component $x_i(t)$ represents the concentration of the i^{th} gene product of the system, for example proteins at time t. The f_i is usually, highly nonlinear function which represents the regulatory interactions of the n genes on the i^{th} gene.

The above system of equations describes how the temporal derivative of the concentration variables depends on the values of the concentration variables themselves. The equations consist of production and degradation constants for each variable in the system, introduced by Tyson and Othmer [36], which enables the feedback loops to be modeled.

Our model derivation utilizes two modeling approaches (Hill and Michealis-Menten formalisms) which are based on ODE modeling. In Chapter 2, these approaches will be examined by both mathematical derivations and graphical representations. Also, two example models from Synthetic Biology which employ these formalisms will be represented.

1.4 Nonlinear Dynamics and Chaos

Rhythms occur at all levels of biological organization, from unicellular to multicellular organisms, with periods ranging from fractions of a second to years. However, oscillatory behavior does not always possess a simple periodic nature. Thus, both in chemistry and biology, oscillations sometimes present complex patterns of bursting, in which successive trains of high-frequency spikes are separated at regular intervals by phases of quiescence. Yet another mode of complex oscillatory behavior is characterized by its aperiodic nature and sensitivity to initial conditions. Such chaotic oscillations have been observed in chemical reactions ([37], [38]) and in a variety of biological contexts ([39], [40], [41]).

So, complex dynamical behavior on a network can be found in a variety of biological networks, such as GRNs, neural networks and the food web. Such systems share a common characteristic: the observed dynamics are robust against disturbance introduced in the dynamics, as well as against disturbance in the network. For example, gene expression patterns obtained through transcription/translation regulations are kept stable in spite of extrinsic noise (i.e. perturbations in dynamics) and mutations (i.e. perturbations in a network).

It has long been hypothesized that living systems favor the edge of chaos, where stability and chaoticity coexist. Originally, Kauffman [42] introduced the Boolean network model as a model of a GRN, and proposed the hypothesis that living systems prefer the edge of chaos because it allows systems to have complex behaviors.

Chaotic motifs are minimal structures with the simplest interactions that can generate chaos. In this section, we will analyse a chaotic motif as a simple three-variable biochemical system, which is also introduced in [1].

1.4.1 Bifurcation

Bifurcation theory, tells us how the generic properties of a dynamical system depend on parameter values. ([43], [44]). The behaviour of a regulatory network is characterized by the attractors of its vector field in a multidimensional state space that we cannot visualize. The nature of these attractors is determined by parameter values that are chosen from a multidimensional *parameter space* that might be extremely large. Fortunately, bifurcation theory predicts that there is only a limited number of ways in which these attractors might transform as we move continuously through parameter space.

These transformations can be visualized by choosing a single variable as being representative of all the interacting proteins in the network, and a single parameter as representative of all the rate constants that are involved in these reactions. By plotting the variable against the parameter (a bifurcation diagram), we get a visual representation of the behaviour of the dynamical system in dependence on its parameters.

Figure 1.9 shows a simple dynamical system undergoing a series of bifurcations leading to a chaotic state.

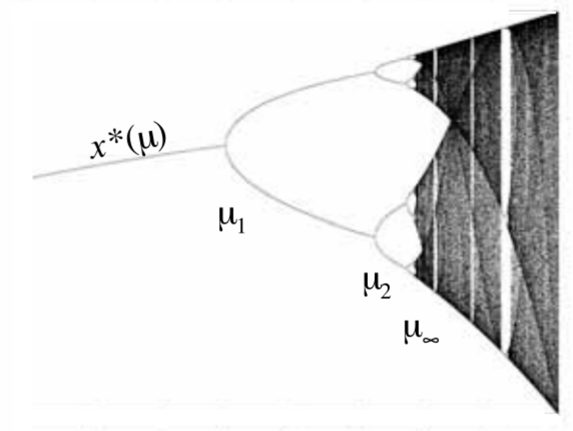


Figure 1.9: The attractor as a function of μ for the Feigenbaum logistic map $f(x) = 4\mu x(1-x)$. For small $\mu < \mu_1$, repeatedly iterating f converges to a fixed-point $x^*(\mu)$. As μ is raised past μ_1 , the map converges into a two-cycle; then a four-cycle at μ_2, an eight-cycle at μ_3... These period-doubling bifurcations converge geometrically: $\mu_\infty - \mu_n \propto \delta^{-n}$ where $\delta = 4.669201609102990...$ is a universal constant. At μ_∞ the system goes chaotic. [45]

1.4.2 Period-Doubling

The appearance of aperiodic oscillations beyond a point of accumulation of a cascade of period-doubling bifurcations is one of the best-known scenarios for the emergence of chaos [46]. The limit cycle undergoes a series of bifurcations through which the oscillations successively acquire $2, 4, 8, 16, ...2^n$ distinct maxima per period; this cascade of period-doubling bifurcations leads to aperiodic oscillations [43].

In the Section 4.3, we will investigate period-doubling bifurcations and corresponding trajectories of a 3-gene network.

2. MODEL DEVELOPMENT

2.1 Mathematical Modeling of Gene Regulation

2.1.1 Michaelis Menten Kinetics of Transcriptional Activation

A regulator can reversibly bind the binding site D of the gene Y (denoted D_0 if unbound and D_1 if bound). The binding/unbinding rates are denoted by k_1 and k_{-1}. Only when activated by the regulator X, the transcription of gene Y can start. The transcription is ensured by the RNA polymerase, P. In a second step, Y-mRNA is translated into Y protein. The transcription/translation rate is denoted by k_t .

The reaction steps for these cases are the following:

$$\begin{gathered} \xrightarrow{k_s} X \xrightarrow{k_d} \\ X + D_0 \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} D_1 \\ D_1 + P \xrightarrow{k_t} D_1 + P + nY \end{gathered} \tag{2.1}$$

In this scheme, we can distinguish several time scales (fast vs slow reactions): The binding/unbinding of the regulatory protein to DNA can occur several times by second, while processes like protein synthesis and gene transcription last over several minutes. Thus, to simplify, the transcription of gene Y and the translation of Y-mRNA can be reduced into a single step. So, the kinetics of the above reaction scheme can be written as:

$$\begin{aligned}\frac{dX}{dt} &= k_s - k_1 D_0 X + k_{-1} D_1 - k_d X \\ \frac{dD_1}{dt} &= k_1 D_0 X - k_{-1} D_1 \\ \frac{dY}{dt} &= n k_t P D_1\end{aligned} \tag{2.2}$$

Because of the fast binding-unbinding rate (k_1 and k_{-1} are high), we can apply the quasi-steady state assumption for the binding/unbinding of the regulator X:

$$\frac{dD_1}{dt} = 0 \tag{2.3}$$

That leads to:

$$k_1 D_0 X = k_{-1} D_1 \tag{2.4}$$

Defining $D_T = D_0 + D_1$ the total concentration of binding sites, we find:

$$k_1 D_T X = (k_1 X + k_{-1}) D_1 \tag{2.5}$$

$$D_1 = \frac{k_1 D_T X}{k_1 X + k_{-1}} = \frac{D_T X}{K_1 + X} \tag{2.6}$$

where K_1 is the dissociation constant, $K_1 = \frac{k_1}{k_{-1}}$. The larger the dissociation constant, the higher the rate of dissociation of complex D_1. So, we find:

$$\frac{dY}{dt} = n k_t P \frac{D_T X}{K_1 + X} \tag{2.7}$$

Therefore, the rate of transcription in the case of an activation by an activator X can be expressed as:

$$V \sim v_{max} \frac{X}{K_1 + X} \tag{2.8}$$

where v_{max} is a limiting rate, K_1 is an enzyme specific constant and X is the substrate concentration. The term $\frac{X}{K_1 + X}$ can be interpreted as the probability of the promoter to be active (i.e. bound to X).

2.1.2 Michaelis Menten Kinetics of Transcriptional Inhibition

We can derive the transcription rate in the case where X acts as a repressor in a similar way.

Assuming the following reaction scheme

$$\begin{gathered} \xrightarrow{k_s} X \xrightarrow{k_d} \\ X + D_0 \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} D_1 \\ D_0 + P \xrightarrow{k_t} D_0 + P + nY \end{gathered} \tag{2.9}$$

with the quasi-steady state assumption

$$\frac{dD_0}{dt} = 0 \tag{2.10}$$

we find

$$V \sim v_{max} \frac{K_1}{K_1 + X} \tag{2.11}$$

where v_{max} is a limiting rate, K_1 is an enzyme specific constant and X is the substrate concentration. The term $\frac{K_1}{K_1 + X}$ can be interpreted as the probability that the promoter is active, i.e. not bound to the repressor X.

2.1.3 Combining Transcriptional Activation and Inhibition

Many genes are regulated by more than one TF. The combined effects of these regulators can be described by a “multi-dimentional transcription function" [18]. A simple case is in which a gene is regulated by an activator X and a repressor Y. A common situation is that the activator and the repressor bind the promoter independently on two different sites.

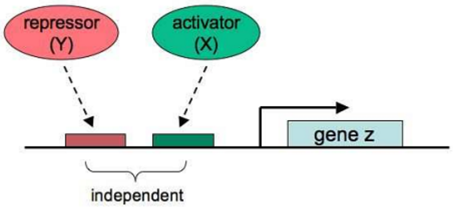


Figure 2.1: Gene regulation by several TFs.

Thus, there are four binding states of promoter D: D, DX, DY, DXY, where DXY means that both X and Y are bound to the promoter. Transcription occurs mainly from the state DX in which the activator but not the repressor is bound.

The probability that X is bound is given by the Michaelis-Menten function (see section 2.1.1):

$$Prob(X\ bound) = \frac{X}{K_1 + X} = \frac{X/K_1}{1 + X/K_1} \tag{2.12}$$

The probability that Y is not bound is given by the Michaelis-Menten function (see section 2.1.2):

$$Prob(Y\ not\ bound) = 1 - \frac{Y}{K_2 + Y} = \frac{1}{1 + Y/K_2} \tag{2.13}$$

Since the two binding events are independent, the probability that the promoter D is bound to X and not to Y is given by the product of the two probabilities:

$$\begin{aligned}
Prob(X\ bound\ and\ Y\ not\ bound) &= Prob(X\ bound) \cdot Prob(Y\ not\ bound) \\
&= \frac{X/K_1}{1 + X/K_1} \cdot \frac{1}{1 + Y/K_2} \\
&= \frac{X/K_1}{1 + X/K_1 + Y/K_2 + XY/K_1K_2}
\end{aligned} \tag{2.14}$$

and the output promoter activity is given by the production rate v_{max} times the probability:

$$v = v_{max} \cdot \frac{X/K_1}{1 + X/K_1 + Y/K_2 + XY/K_1K_2} \tag{2.15}$$

This results in an *“X AND NOT Y”* transcription function.

2.1.4 Transcriptional Regulation with Multiple Binding Sites (Hill Kinetics)

As portrayed in the previous section, most proteins (genes) and enzymes found in living organisms are composed of more than one, mostly identical, binding sites. If one ligand binds to such a macromolecule, a conformational change alters the binding characteristics at all binding sites and a consecutive binding process is triggered. Such mechanisms, which assume that one ligand supports binding of others is called *cooperativity.*

As shown in Figure 2.2, two types of TFs quantitatively control the expression rate by their present concentration. Inhibitors, here symbolised by $I_1, ..., I_p$, repress gene expression while activators $A_1, ..., A_n$ cause the opposite amplifying effect. By coupling gene regulatory units we obtain GRNs, where gene products can act as TFs for other genes within the network.

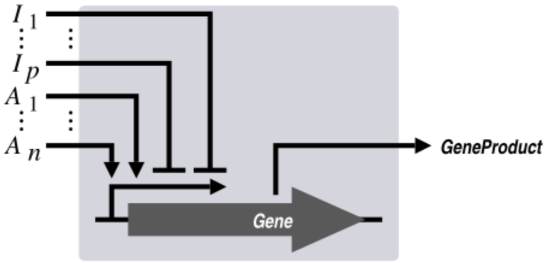


Figure 2.2: Gene regulatory unit [47]. Repetitive expression of a *gene* leads to generation of a specific *gene product.*

Even at high ligand concentrations, binding will be initiated by occupation of one binding site by the first ligand, followed by binding of the second one, and so on [48]. This is formulated by the reaction sequence with n ligand binding sites:

$$\begin{aligned} E + A &\rightleftarrows EA \\ EA + A &\rightleftarrows EA_2 \\ EA_2 + A &\rightleftarrows EA_3 \\ &\vdots \\ EA_{n-1} + A &\rightleftarrows EA_n \end{aligned}$$

So, the overall reaction sequence is formulated by:

$$E + nA \rightleftarrows EA_n$$

For the derivation of the general binding equation, a rate function r is defined as the quotient from the portion of bound ligand to the total amount of the macromolecule:

$$r = \frac{[A]_{bound}}{[E]_0} = \frac{[EA] + 2[EA_2] + 3[EA_3] + \ldots + n[EA_n]}{[E] + [EA] + [EA_2] + [EA_3] + \ldots + [EA_n]} \tag{2.16}$$

The concentrations of the individual macromolecule forms are replaced by the dissociation constants:

$$\begin{aligned}
K_1 =& \frac{[E][A]}{[EA]}; \qquad [EA] = \frac{[E][A]}{K_1} \\
K_2 =& \frac{[EA][A]}{[EA_2]}; \qquad [EA_2] = \frac{[EA][A]}{K_2} = \frac{[E][A]^2}{K_1K_2} \\
K_3 =& \frac{[EA_2][A]}{[EA_3]}; \qquad [EA_3] = \frac{[EA_2][A]}{K_3} = \frac{[E][A]^3}{K_1K_2K_3} \\
&\vdots \qquad\qquad\qquad\qquad \vdots \\
K_n =& \frac{[EA_{n-1}][A]}{[EA_n]}; \qquad [EA_n] = \frac{[EA_{n-1}][A]}{K_n} = \frac{[E][A]^n}{K_1K_2K_3\ldots K_n}
\end{aligned}$$

Thus, the rate function evolves as:

$$r = \frac{\dfrac{[A]}{K_1} + \dfrac{2[A]^2}{K_1K_2} + \dfrac{3[A]^3}{K_1K_2K_3} + \cdots + \dfrac{n[A]^n}{K_1K_2K_3\ldots K_n}}{1 + \dfrac{[A]}{K_1} + \dfrac{[A]^2}{K_1K_2} + \dfrac{[A]^3}{K_1K_2K_3} + \cdots + \dfrac{[A]^n}{K_1K_2K_3\ldots K_n}} \tag{2.17}$$

It is known [49] that the first ligand binds with very low affinity (i.e. large K_1), and all subsequent ligands binds with an increasing affinity ($K_1 > K_2 > \ldots > K_n$). Thus, by approximation, all terms except having K_n in the denominator goes to zero.

$$r = \frac{\dfrac{n[A]^n}{K_1K_2K_3\ldots K_n}}{1 + \dfrac{[A]^n}{K_1K_2K_3\ldots K_n}} \tag{2.18}$$

Calling $K^n = K_1K_2K_3\ldots K_n$, the equation becomes:

$$r = \frac{\dfrac{n[A]^n}{K^n}}{1 + \dfrac{[A]^n}{K^n}} \tag{2.19}$$

Therefore, the rate at which the product X is formed, is then approximated by:

$$\frac{dX}{dT} = v_{max}\frac{[X]^n}{K^n + [X]^n} \tag{2.20}$$

where v_{max} is a limiting rate, K is an enzyme specific constant, n is the Hill coefficient and X is the substrate concentration.

For the cooperative inhibition case, the probability that “Y is not bound” can be found similar to section 2.1.3 , so that the Hill function for the product Y can be found by [50, p. 291]

$$\frac{dY}{dT} = v_{max}(1 - \frac{[Y]^n}{K^n + [Y]^n}) = v_{max}\frac{K^n}{K^n + [Y]^n} \tag{2.21}$$

For further information, mechanistic interpretations and experimental evidence for Hill functions are discussed in [51].

2.1.5 Example of the Biological Switch

One of the simplest example of model of GRNs using differential equations is the so called *biological switch.* (Figure 2.3)

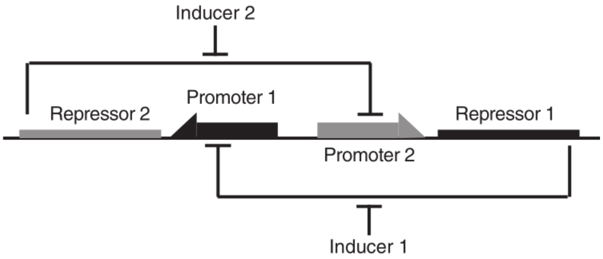


Figure 2.3: Toggle switch design [21]. *Repressor 1* inhibits transcription from *Promoter 1* and is induced by *Inducer 1*. *Repressor 2* inhibits transcription from *Promoter 2* and is induced by *Inducer 2*.

The model consists of two genes X_1 and X_2. We assume the dynamics of the genes are similar: inhibition by the other gene (described by a Hill function) and decreasing of the level of concentration of its protein due to degradation. These assumptions lead to the following ODE system:

$$\begin{aligned}\frac{dX_1}{dt} =& k_1 \frac{{\theta_2}^n}{{X_2}^n + {\theta_2}^n} - \gamma_1 X_1 \\ \frac{dX_2}{dt} =& k_2 \frac{{\theta_1}^n}{{X_1}^n + {\theta_1}^n} - \gamma_2 X_2\end{aligned} \tag{2.22}$$

Many biological systems respond switch-like to changes in input. Depending on the initial conditions, this system converges to one equilibrium point corresponding to a maximal level of expression of X_2 and about no expression of X_1, or the level of expression of X_1 is maximal and there is about no expression of X_2.

2.1.6 Example of the Goodwin's Model

The second example is a biochemical oscillator which was invented by Brian Goodwin [52]. The simplest Goodwin model consists three variables (Figure 2.4). Many versions have been developed after his first model [53].

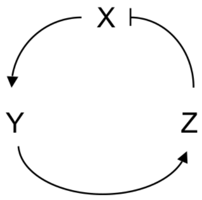


Figure 2.4: Goodwin's oscillator involves a single gene that inhibits itself, based on a delayed negative feedback loop.

These three variables model may be written by the following system of differential equations:

$$\begin{aligned}\frac{dX}{dt} =& k_1 \frac{K^n}{Z^n + K^n} - \gamma_1 X\\ \frac{dY}{dt} =& k_2 X - \gamma_2 Y\\ \frac{dZ}{dt} =& k_3 Y - \gamma_3 Z\end{aligned} \tag{2.23}$$

The Goodwin model was originally proposed to model oscillatory processes in enzymatic control processes. Due to its generic nature, it was generalized in many forms with different contexts (e.g. gene regulatory processes).

2.2 A Graphical Representation of Cooperative Binding Mechanism

Regulation of transcription is a dynamic process involving repeated association and dissociation of TFs to operator sites on the DNA. As previously stated, the times involved in TF binding to operator sites (less than a minute) are often faster than durations of protein production (transcription and translation, typically a minute or more). Therefore, most models assume that association-dissociation occurs so fast that TF-operator complexes are considered to be in quasi-equilibrium. In other words, we can assume that, the fraction of time the TF spends bound to the operator adjusts instantaneously to changes in concentrations of the involved proteins and DNA.

Under this approximation, the bound fraction is given by the following function of the regulator (the TF) concentration, X, and the dissociation constant, K.

$$\textit{bound fraction} \equiv \frac{X/K}{1 + X/K} \tag{2.24}$$

This is similar to the Michaelis-Menten description of enzyme kinetics, with X correspondent to substrate concentration, and K correspondent to the Michaelis constant describing substrate enzyme binding. Just as the Michaelis-Menten formulation can be extended to incorporate cooperative effects, so can the above formula be generalized by adding a Hill coefficient n:

$$\textit{bound fraction} \equiv \frac{(X/K)^n}{1 + (X/K)^n} \tag{2.25}$$

Cooperative binding of multiple TFs, is represented by Hill coefficients $h > 1$. The above equation can be easily extended to describe different TFs competing for multiple operator sites [54].

A common type of link in a regulatory network (RN) represents a TF activating or repressing a promoter. If the TF activates the promoter, its activity is proportional to the bound fraction from Eq. 2.25:

$$\textit{Activity} \sim \frac{(X/K)^n}{1 + (X/K)^n} \tag{2.26}$$

Whereas, if the TF represses the promoter, its activity equals:

$$\textit{Activity} \sim \frac{1}{1 + (X/K)^n} \tag{2.27}$$

A large Hill coefficient makes activation more sensitive to variations in X when X is approximately equal to K, approaching a step function when n becomes very large. Notice that, in Eq. 2.26 and Eq. 2.27 we have not explicitly modeled transcription and translation by using a separate equation for the mRNA concentration, which is reasonable whenever mRNA turnover is faster than protein turnover.

Given the simplifications made above, the dynamics of the concentration of a protein, X, produced from a gene repressed by a TF can then be modeled in the following form:

$$\frac{dX}{dT} = \textit{capacity} \; . \frac{1}{1 + (^{X}/_{K})^{n}} - \textit{degradation} \tag{2.28}$$

where we used v_{max} term before to stand for the capacity. "*capacity*" accounts for cases in which repression cannot reach *100 %* capacity, sets the maximum production rate of the protein. The positive terms model production, whereas the last term models degradation of the protein.

There is a virtual demonstration of the cooperative binding mechanism, that shows how the Michaelis functions approach above can be reexamined utilizing partition functions (Figure 2.5). The essence of the framework is to compute the ratio of transcriptionally active promoter states to the sum of all states, active and inert. This ratio depends on variables including the binding sites present in the promoter, the concentrations of proteins that bind these sequences, and the affinities of the TF-DNA interactions.

For any particular system, construction of a regulatory function requires three components: (1) a list of all states, (2) the binding constant for each state, and (3) the information for whether each state is capable of transcribing or not.

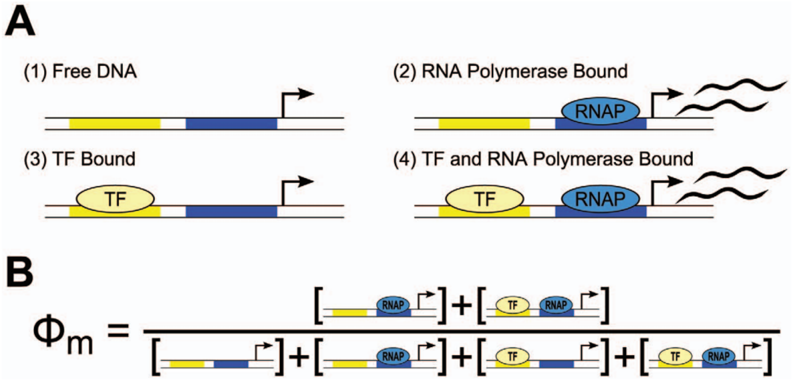


Figure 2.5: Graphical representation of TF binding. **(A)** States 1 and 3 indicate where transcription is inactive, and states 2 and 4 indicate where transcription is active. **(B)** The ϕ_m function is composed of sum of the concentrations of transcriptionally active states divided by the sum of the concentrations of all possible states. [55]

In Figure 2.5, state-1 at the denominator corresponds to the reference state where DNA has nothing bound. State-2 has RNAP bound by itself, state-3 has TF bound by itself, and state-4 has both TF and RNAP bound. Simply by writing these states, we are already specifying the architecture of our system. States at numerator indicates that which of these states are capable of transcribing.

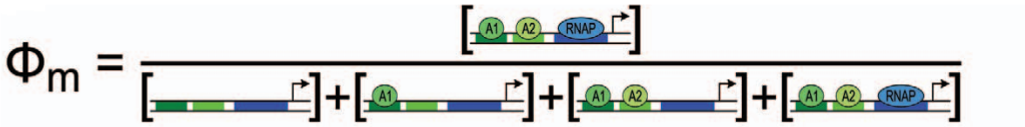


Figure 2.6: Graphical representations of competitive binding. Sequential binding model. The ovals correspond to RNAP (blue), activator A1 (dark green), and activator A2 (light green). [55]

In Figure 2.6 there are two binding sites for two activators and their correspondent function is:

$$\phi_m = \frac{\frac{A_1.A_2}{K_{A1}.K_{A2}}}{1 + \frac{A_1}{K_{A1}} + \frac{A_1.A_2}{K_{A1}.K_{A2}}} \tag{2.29}$$

Similarly, the function considering repressors binding has no term on the numerator:

$$\phi_m = \frac{1}{1 + \frac{R_1}{K_{A1}} + \frac{R_1.R_2}{K_{A1}.K_{A2}}} \tag{2.30}$$

Because of the saturation condition $K_{A1} > K_{A1}.K_{A2}$, we can eliminate the term divided by K_{A1}. So, we can generalize these functions and simply writing in the following form:

$$\phi_m = \frac{1}{1 + (\sum\limits_{j=1}^{N} {}^{R_j}/_{K_j})^n} \tag{2.31}$$

where n stands for the Hill coefficient.

Likewise, for the activator binding, the simplified form is:

$$\phi_m = \frac{(\sum\limits_{j=1}^{N} {}^{A_j}/_{K_j})^n}{1 + (\sum\limits_{j=1}^{N} {}^{A_j}/_{K_j})^n} \tag{2.32}$$

2.3 Modular Michaelis Functions as Partition Functions

As previously mentioned, when there are several TFs that jointly regulate gene i, its expression is determined by a cis-regulatory input function that integrates regulation by TFs. Likewise, in our approach, we assume such multiple TFs act competitively to regulate a gene. Therefore, the concentrations of these regulatory proteins are additively affecting gene expression, so that the production rate is written in the summation form.

On the other hand, to increase the rate of production, products binding to a regulatory region, can work either additively or multiplicatively. In the latter mode, presence of a product binding to a multiplicative promoter is necessary to activate the synthesis and amplifies it proportionally to the activation of such promoter. This input function has a form of

$$\frac{dX}{dT} = \prod_{i=1}^{m} A_i \prod_{j=1}^{n} R_j \tag{2.33}$$

where A is an activatory and R is an inhibitory regulatory protein.

Including the multiplicative binding, an activator-inhibitor model [56, p. 244] can be constructed as

$$\frac{dX_i}{dt} = V_i \frac{(\sum_{j=1}^{N} {}^{A_j}/_{K_{ij}})^n}{1 + (\sum_{j=1}^{N} {}^{A_j}/_{K_{ij}})^n} . \frac{1}{1 + (\sum_{j=1}^{N} {}^{R_j}/_{K_{ij}})^n} - \gamma_i X_i \tag{2.34}$$

where R reduces the overall speed of the catalysis independently of A.

This ODE model includes a term for activators, in addition to the repressor included ODE model studied in [57].

2.4 Non-dimensionalisation

For our study, we have chosen a simplified model of a genetic network having activatory and inhibitory interactions between genes. The activity of any gene can be regulated by itself or other genes in the network. We will consider connected networks of N genes with M links, where N and M having fixed values.

The mathematical background and the steps of derivation of the ODE are described in previous sections. Accordingly, if we assume that the decay rates are the same for all proteins, $\gamma_i = \gamma$, the i^{th} protein undergoes degradation with rate constant γ_i, the change of concentration for this protein is given by the following ODE.

$$\frac{dU_i}{dt} = V_i \frac{(\sum_{j=1}^{N} {}^{U_j^A}/{}_{K_{ij}})^n}{1 + (\sum_{j=1}^{N} {}^{U_j^A}/{}_{K_{ij}})^n} \cdot \frac{1}{1 + (\sum_{j=1}^{N} {}^{U_j^R}/{}_{K_{ij}})^n} - \gamma_i U_i \tag{2.35}$$

where K_{ij} specifies the strength of the effect of the regulatory factor U_j^A (or U_j^R) on the production rate of protein U_i, $i = 1, 2, ..., N$ denotes the gene index, U_j^A and U_j^R stand for the molar concentrations of j^{th} activator and repressor proteins respectively.

The structure of a regulatory network having N genes is specified by an adjacency matrix A_{ij}, which is defined in such a way that $A_{ij} = 1$, if an activatory interaction from the gene j to i exists, and $A_{ij} = 0$ otherwise. For example, the adjacency matrix can be described by

$$A = \begin{bmatrix} 1 & 0 & \dots & 1 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \dots & 1 \end{bmatrix}$$

So including the activatory (A_{ij}) and inhibitory (R_{ij}) adjacency matrices, the ODE form for such a regulatory network becames:

$$\frac{dU_i}{dt} = V_i \frac{(\sum_{j=1}^{N} {}^{A_{ij}U_j}/{}_{K_{ij}})^n}{1 + (\sum_{j=1}^{N} {}^{A_{ij}U_j}/{}_{K_{ij}})^n} \cdot \frac{1}{1 + (\sum_{j=1}^{N} {}^{R_{ij}U_j}/{}_{K_{ij}})^n} - \gamma_i U_i \tag{2.36}$$

Instead of t, introducing dimensionless time constant $\tau = \gamma t$ by measuring time in the units of γ^{-1}, we have $d\tau = \gamma dt$. Furthermore, inserting the dimensionless concentrations $u_i = \frac{\gamma U_i}{V_i}$ and , the activity of the regulatory network gets described by the following set of ODE:

$$\frac{du_i}{d\tau} = \frac{1}{1+(\sum_j R_{ij}\frac{u_j V_j}{K_{ij}\gamma})^n} \frac{(\sum_j A_{ij}\frac{u_j V_j}{K_{ij}\gamma})^n}{1+(\sum_j A_{ij}\frac{u_j V_j}{K_{ij}\gamma})^n} - u_i \tag{2.37}$$

Also, if we assume that ϕ's are the same for all proteins, $\phi_{ij} = \phi$; then $\phi_{ij} = \frac{V_j}{\gamma K_{ij}}$ is simplified to $\phi = \frac{V}{\gamma K}$:

$$\frac{du_i}{dt} = \frac{(\phi \sum\limits_{j=1}^{N} A_{ij}u_j)^n}{1+(\phi \sum\limits_{j=1}^{N} A_{ij}u_j)^n} \cdot \frac{1}{1+(\phi \sum\limits_{j=1}^{N} R_{ij}u_j)^n} - u_i \qquad (i = 1, 2, ..., N) \tag{2.38}$$

2.5 Parameter Specification

n : The Hill coefficient n, has been described more appropriately as an "interaction coefficient" that provides only a minimum estimate of the number of binding sites involved [58]. For example, in the case of hemoglobin, in which four oxygen molecules are known to bind with a high degree of positive cooperativity, the measured Hill coefficient ranges from 1.7 to 3.2 [59]. In our model, it is taken as $n = 3$, as in [57].

$\boldsymbol{\gamma}$: For the conversion of our dimensionless time parameter, τ, into real time scale, we referred the experimentally constructed repressilator model [22]. In this publication, the protein half-life is given as nearly *10 mins.* Using the exponential decay formula

$$N(t) = N_0 e^{\gamma t}$$

we can derive the γ constant:

$$\gamma = \frac{ln(2)}{t_{1/2}} = \frac{ln(2)}{10\ mins} = 0.069$$

Our dimensionless time constant was defined as $\tau = t\gamma$. For a realistic transformation, we can use the γ value to find the real period as

$$t = \frac{\tau}{\gamma} = \frac{10.62}{0.069} \approx 154 \; mins$$

which is very close to the period of the repressilator which is found to be approximately *150 mins* in [22].

ϕ : According to [60], by an approximation the value of the parameter ϕ is chosen as 100. That value falls into median of the interval stated in [1], where the parameter ϕ renamed as K:

$$0.005 < K < 0.30$$

So, instead of $\phi\prime \equiv \frac{V}{\gamma}$, if we use $\phi \equiv \frac{V}{\gamma K}$; we get the framework stated in [1]. Thus, the corresponding interval for ϕ is

$$3.33 < \phi < 200 \; .$$

3. THE EVOLUTIONARY OPTIMIZATION METHOD

In recent decades, there is a great deal of study on developing and testing models of GRNs. As GRN models become more developed, a greater understanding of the particular “wiring”s of regulatory dynamics, allows us to analyze how these GRNs evolved. Simulations of GRN evolution, benefit from optimization techniques of evolutionary computations (ECs). In modeling a dynamical system represented by GRNs, it is possible to make an *in silico* simulation of the evolution of a gene expression over time. In this way, one can study how the system behaves when it is slightly changed to reflect gene mutations inside a cell. In the strict sense, *in silico* evolution mimics the process throughout a heuristic search in which a system is iteratively modified (mutation process) and evaluated for its performance towards a pre-defined target (fitness computation). Therefore, depending on the size and complexity of the network, a large-scale and high performance evolutionary computation is required over many generations.

In one of the pioneering studies, they have focused on the use of evolutionary optimization algorithms to engineer synthetic regulatory circuits with specified functionalities [61]. Also, a paper from last decade [62], describes how an evolutionary approach can be used to generate functional modules (oscillators, bistable switches, homeostatic systems and frequency filters), with the goal of building a library of modules categorized according to their function. Furthermore, the review [63] highlights several examples of models that have been designed using such procedures, together with different objective functions to select for the proper behavior. It discusses the basic principles of evolutionary approaches and how they can be applied to engineer synthetic networks.

In addition, in a recent study, they have utilized an evolutionary algorithm on a discrete stochastic modeling approach. This study proposes an automated design of cell models by combining different modules [64]. At the same year, another publication discusses several evolutionary algorithm approaches for quantitative GRN modeling. These algorithms of interest have been applied both synthetic and real gene expression data, so this study offers a comprehensive comparison of approaches [65].

To sum up, artificial GRNs have been instrumental in elucidating basic principles that govern the dynamics and consequences of randomness in the gene expression of naturally occurring GRNs. The design of computational circuits, helps us to infer inherent evolutionary fault tolerance and robustness of these GRNs.

3.1 Evolutionary Algorithms

Evolutionary algorithms are structurally very simple and they work in rounds that are called *generations.* They operate on some *search space*, where points are assigned some quality via a function f. In the context of optimization, f is usually called *fitness function.* ([66])

An evolutionary algorithm operates on a collection of points from the search space, called a *population* P. The members of the population, i.e., some points in the search space, are called *individuals.* Choosing the first population P_0 in the beginning is called *initialization.* Then, usually for each member of the population its fitness function is computed and stored. Then some random *variation* is applied to the initial population where small changes are more likely than large changes. These variation operators are called *mutation.*

After creating the offspring population, there is *selection for replacement* based on the reason that the size of the population is not changed during the whole run. Thus, the new population P_{t+1} is selected from the old population P_t and the newly generated offspring. Selection for reproduction in preferring individuals with smaller fitness and the fitness value is either available or will be computed on the run. After the new generation P_{t+1} is produced (and the old population P_t is discarded), it is checked whether some *termination criterion* is met. If so, some output is produced and the algorithm terminates.

Evolutionary algorithms are a kind of general randomized search heuristics. Empirical optimization methods, e.g. simulated annealing [67], do not allow to find an exact solution, but they are efficient in identifying an approximate solution which is close to the target. In network design, models are typically based on a network whose topology is evolving towards the optimal target topology that performs prescribed functions. Most used among such models is the evolutionary optimisation, a version of Monte Carlo simulated annealing algorithm [68]. Here we will refer to the random local search and Metropolis algorithm, which is a type of Monte Carlo simulated annealing algorithm.

Random local search: The search starts with some $x \epsilon \{0,1\}_n$ chosen uniformly at random. In each step another point $y \epsilon N(x)$ is chosen uniformly at random, where $N(x)$ denotes some neighborhood of x. Then, y replaces x if $f(y) \geq f(x)$ holds.

Metropolis algorithm: The search starts with some $x \epsilon \{0,1\}_n$ chosen uniformly at random. In each step another point $y \epsilon N(x)$ is chosen uniformly at random, where $N(x)$ denotes some neighborhood of x. Then, y replaces x with probability $min\{1,\ e^{f(y)-f(x)}/T\}$.

As for random local search, the most common neighborhood comprises just of the Hamming neighbors of x. The parameter T is called temperature. It is fixed and held constant during the complete run. The term $min\{1,\ e^{f(y)-f(x)}/T\}$ equals 1 if $f(y) \geq f(x)$ holds. So, improvements in fitness are always accepted. But for $f(y) < f(x)$, the two search heuristics differ.

While such a move from x to y is never done with random local search, it can be done in the Metropolis algorithm. The probability, however, depends on the parameter T and the difference in fitness values $f(y)-f(x)$. With increasing difference, the probability for such a step decreases exponentially. The selection mechanism helps to avoid from getting stuck in local optima.

3.2 The Evolutionary Algorithm Used In The Source Code

The aim of the evolutionary optimization is to find networks which can generate oscillations with arbitrary prescribed periods. The parameters of genetic interactions should be the same for all such networks, and only their architectures may differ. The search for a network architecture which is able to generate persistent oscillations with a given period can be viewed as an *optimization* process.

In this process, through repeated rewiring, a network yielding regular oscillations with a period sufficiently close to the target can be identified. Our evolutionary optimization process is utilizing the Metropolis algorithm which is in stochastic Monte Carlo simulations [57]. It consists of a sequence of iteration steps. In each step, a structural mutation is applied to a network and the change in its performance (the cost function) is determined. Then the decision, whether to accept or neglect the mutation, is made.

A dynamic network generates some time-dependent output signal $u_1(t)$, not necessarily periodic. A certain sequence of events can be associated with this signal, and time intervals between these events can be determined. The definition of an "event" is that, the output signal crosses from below a certain threshold h (In Figure 3.1)

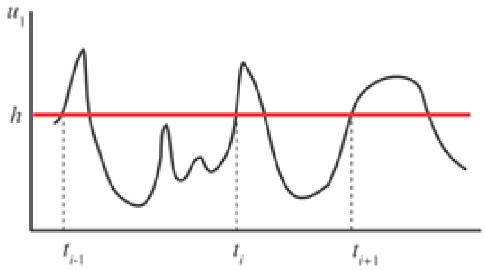


Figure 3.1: Determination of the time series t_i from the output signal u_1 and the threshold h. [57]

Monitoring the output signal for a sufficiently long time and finding the maximum and minimum bounds u_{max} and u_{min} of the signal, the threshold is chosen as $h = (u_{max} + u_{min})/2$. Explicitly, we determine time moments $t_i(i = 1, ..., K)$ at which $u_1(t) = h$ and $du_1/dt > 0$. Using time intervals $\Delta i = t_i - t_{i-1}$ between the events, we compute the average period $T =< \Delta_i >= (1/K)\sum_{i=1}^{K} \Delta_i$ and the variance $\sigma^2 =< \Delta_i^2 > - < \Delta_i >^2$.

If a signal must be periodic, its variance σ should vanish. Moreover, we want that the period of the output signal coincides with some prescribed period T_0. This means, in our optimization problem, the following cost function can be utilized

$$\epsilon = \frac{(T - T_0)^2}{T_0^2} + \frac{\sigma^2}{T^2}$$

The cost function ϵ acquires its minimal possible value of zero when we have a periodic signal with the period T_0.

Our optimization algorithm involves structural "mutations" of networks, which will consist of random rewiring of the links. We randomly choose a link in a given network and delete it. Then, we randomly choose again, which two network nodes should be connected by this link after the mutation. A link, after its relocation, can only connect the nodes which were not linked before that. So, in every mutation, the total number of the links is assumed to be conserved.

To perform optimization, we start with a random network and check its dynamics. If this network turns out to be in a stationary state, we drop it and randomly choose another initial network, until a network with some dynamics is found. Then a mutation (a link rewiring) is applied. By running dynamical simulations of the network before and after the mutation, we determine their average periods and the variances and thus compute the values ϵ and $\epsilon^{'}$ before and after the mutation. We accept it with some probability $p = exp(-\frac{\Delta\epsilon}{\epsilon\mu})$, if $\Delta\epsilon \geq 0$. The iterations should be repeated until the variance σ vanishes and, additionally, until the difference $|T - T_0|$ becomes smaller than some given tolerance threshold ΔT.

The value of the parameter μ is chosen with regard to acceptance ratio. Choosing the acceptance ratio is important for a good numerical performance of the algorithm.

It should be nearly *50 %* for unidimensional problems [69]. So, after a few trials μ is assigned as 0.2, which almost gives that acceptance ratio.

3.2.1 Specified Description of the Simulation Algorithm

The evolutionary optimization algorithm is written in Python programming [70] based on the implementations below.

- **Initialization:** The simulation algorithm starts with the initialization of a randomly generated gene network that controlled by many activators and repressors.
- **Mutation:** Mutation process is based on link rewiring. After generating a mutated network, its dynamics are checked. If it is in a stationary state, it is dropped. The process is repeated until another initial network with some dynamics is found.
- **Selection:** Oscillation profile of the accepted network is evaluated by measuring its average period and variance. Then, the cost function is evaluated based on these values. If it is within the intended period interval $|T - T_0|$, it is accepted. The selection in this case was aimed at obtaining stable oscillations at the imposed interval.

This evolution process is iterated over the course of an optimal time, generated many networks with different architectures, which correspond to different dynamics.

The main idea and pseudocode for Metropolis algorithm can be found at Appendix A.

4. ANALYSIS

4.1 Statistical Analysis of Randomly Generated Networks

For this part, we have written a module script that uses the NetworkX graph generator library [71] which can generate random networks and visualization. Considering the computation time, we decided to investigate considerably small networks including 4 and 5 genes.

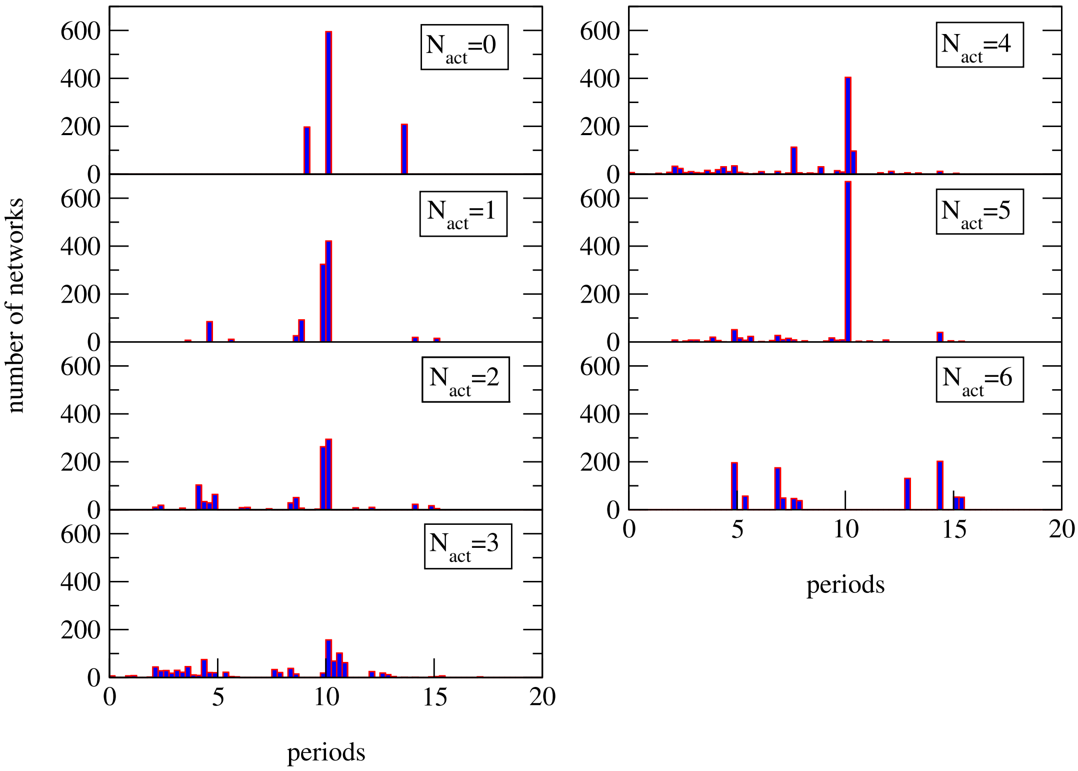


Figure 4.1: Period histograms of randomly generated networks with 4 nodes & 8 links.

For $N_{act} = 5$ case, peaks at the histogram shows that the number of networks which oscillates at period $T = 10.25$ is maximum. If we consider $N_{act} = 5$ case, there is one repressilator motif (3 inhibitory and 5 activatory links). So, these networks are frequently encountered with their structurally stable dynamics. Considering the histograms for the other cases, we see that the number of networks which oscillates at that peak period ($T = 10.25$) diminishes. In addition, histogram for $N_{act} = 6$ becomes uniform and there is no peak period at 10.25 which is seen in all of the other histograms. We can notice that this fact is based on the same reasoning: the repressilator motif is lost because there are only two repressor links.

Table 4.1: Statistics of randomly generated networks with 4 nodes & 8 links

# of activators	percentage of oscillatory networks	first three peak periods
0	6.37	10.25, 13.75, 9.25
1	7.23	10.25, 10.00, 9.00
2	7.00	10.25, 10.00, 4.25
3	6.37	10.25, 10.75, 4.50
4	6.05	10.25, 7.75, 10.50
5	4.44	10.25, 5.00, 14.50
6	0.89	14.50, 5.00, 7.00

In the Table 4.1, there are the first three peak periods for each histogram. The first peak at 10.25 is significant result that arises from a dominant represilator motif hidden in these networks. Crucially, that dynamic is lost when there are only two inhibitory links.

In the same table, as the number of activators increases from 1 to 6, the percentage of oscillatory networks decreases. This implies that, the increasing number of activatory links leads the network dynamics to a major positive feedback which amplifies the perturbation.

Similar to histograms in Figure 4.1, for $N_{act} = 7$ case in Figure 4.2, peaks at the histogram shows that the number of networks which oscillates at period $T = 10.25$ is maximum. Also, histogram for $N_{act} = 8$ becomes uniform and there is no peak period at 10.25 which is seen in all of the other histograms.

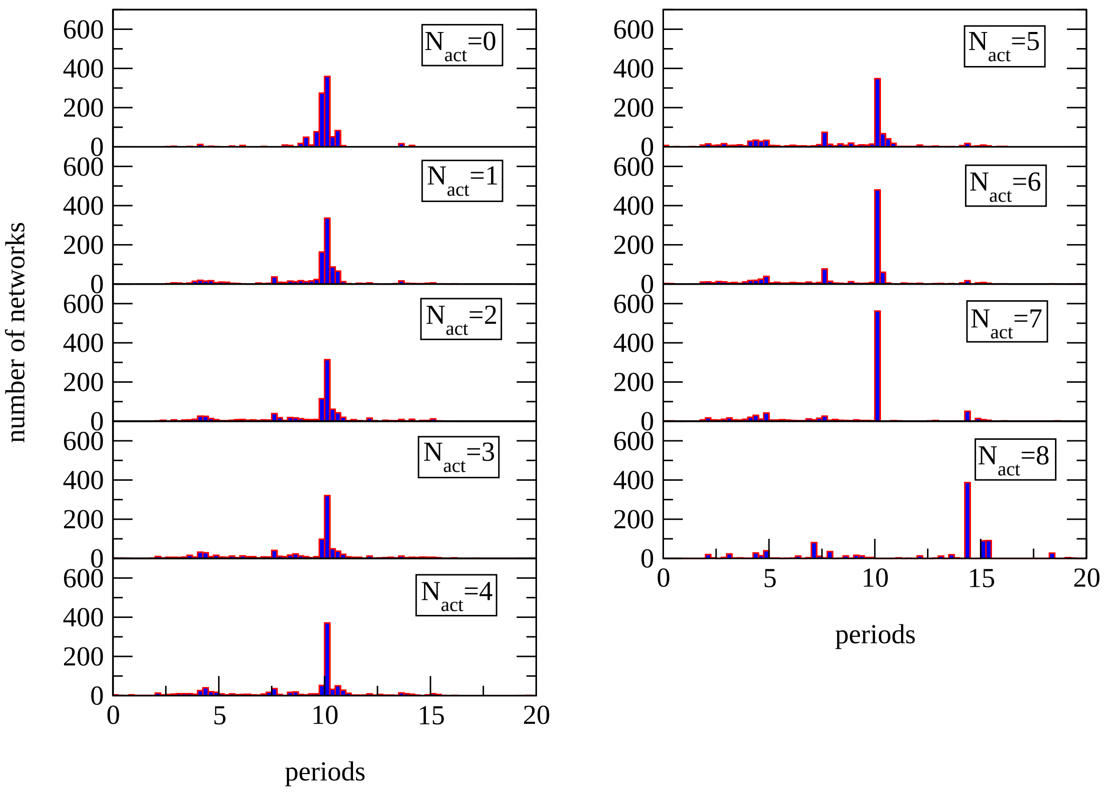


Figure 4.2: Period histograms of randomly generated networks with 5 nodes & 10 links.

Table 4.2: Statistics of randomly generated networks with 5 nodes & 10 links

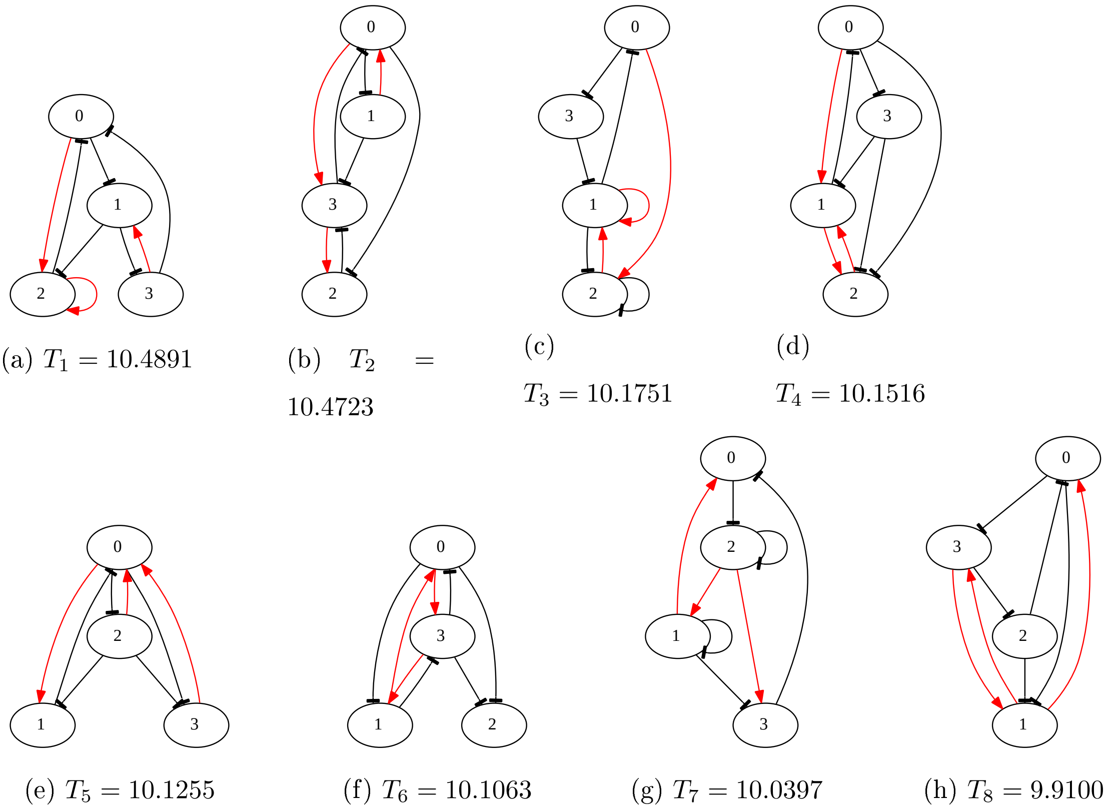
# of activators	percentage of oscillatory networks	first three peak periods
0	6.61	10.25, 10.00, 10.75
1	7.09	10.25, 10.00, 10.50
2	7.28	10.25, 10.00, 10.50
3	6.81	10.25, 10.00, 10.50
4	6.57	10.25, 10.00, 10.75
5	5.58	10.25, 7.75, 10.50
6	4.43	10.25, 7.75, 10.50
7	2.14	10.25, 14.50, 5.00
8	0.50	14.50, 15.50, 7.25

Similar to the Table 4.1, in the Table 4.2, the first peak at 10.25 is a significant result that arises from a dominant represilator motif hidden in these networks. Again, that dynamic is lost when only two inhibitory links are left.

4.2 Evolutionary Analysis of Networks with 4 Genes

We runned the Metropolis algorithm for networks with 4 genes and 8 links, in which 3 of these links are activatory. This structure is chosen based on its oscillatory behaviour within a broader range of period values compared to the other histogram plots (Figure 4.1). In this process, 100 initial networks are simulated under repeated mutations over maximum 1000 iterations to reach the given target periods.

The search for target period 10 with *5%* tolerance, has performed succesfully for all of the initial networks, and it has resulted with 27 non identical networks. Some of these network structures and their respective periods are shown in Figure 4.3.



(a) $T_1 = 10.4891$

(b) $T_2 = 10.4723$

(c) $T_3 = 10.1751$

(d) $T_4 = 10.1516$

(e) $T_5 = 10.1255$

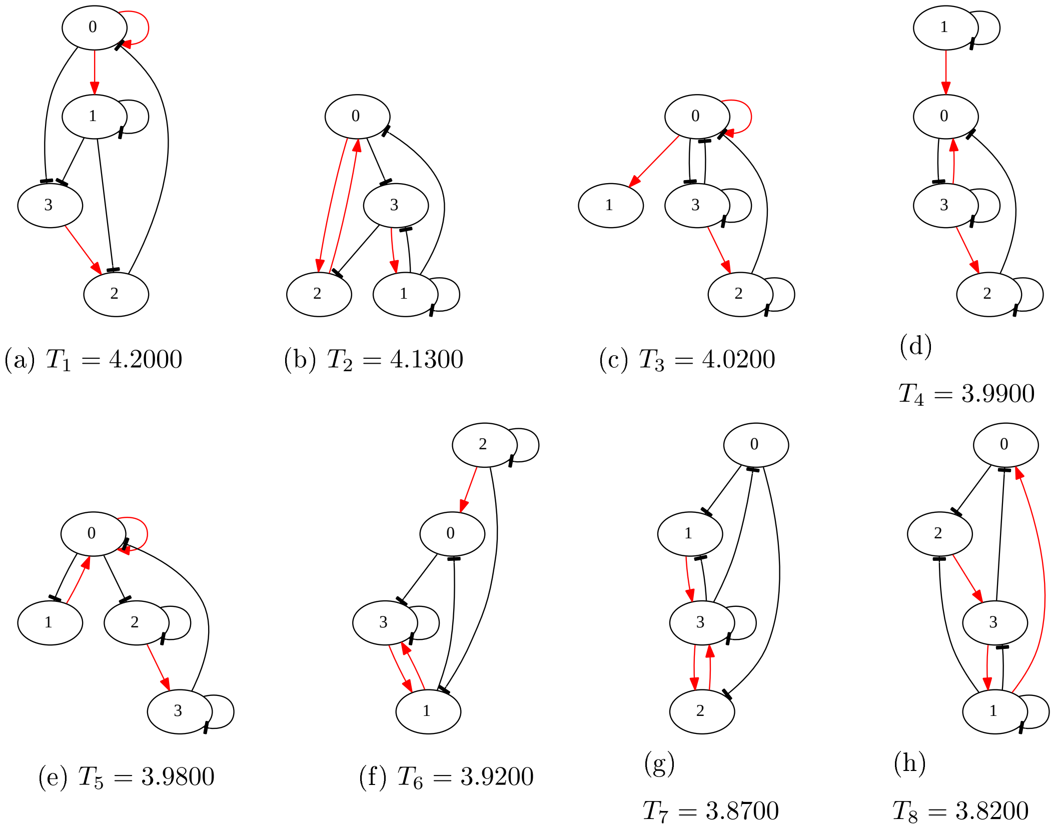
(f) $T_6 = 10.1063$

(g) $T_7 = 10.0397$

(h) $T_8 = 9.9100$

Figure 4.3: Examples from designed networks for target period 10.

Furthermore, the search for target period 4 with *5%* tolerance, has performed succesfully for all of the initial networks, and it has resulted with 15 non identical networks. Some of these network structures and their respective periods are shown in Figure 4.4.



(a) $T_1 = 4.2000$

(b) $T_2 = 4.1300$

(c) $T_3 = 4.0200$

(d) $T_4 = 3.9900$

(e) $T_5 = 3.9800$

(f) $T_6 = 3.9200$

(g) $T_7 = 3.8700$

(h) $T_8 = 3.8200$

Figure 4.4: Examples from designed networks for target period 4.

4.3 Bifurcation Analysis and Limit Cycle Behaviour of a 3-Gene Network

The 3-gene network in Figure 4.5 consists a repressilator network with additional three activator links from 0^{th} node to 1^{st} node, 2^{nd} node to 0^{th} node, and 2^{nd} node to itself. Dynamical behaviour of this network is investigated before in [1] using the same mathematical model including a different parameter name.

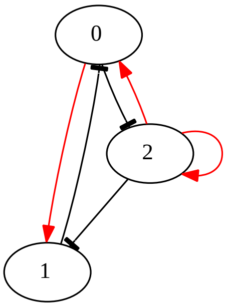


Figure 4.5: Three node gene system with 3 activatory and 3 inhibitory links.

Critical points can be determined from the bifurcation plot (Figure 4.6), in which the system switches to a new behavior with twice the period of the original system. In continuous dynamical systems, when a new *limit cycle* emerges from an existing limit cycle, and the period of the new limit cycle is twice that of the old one.

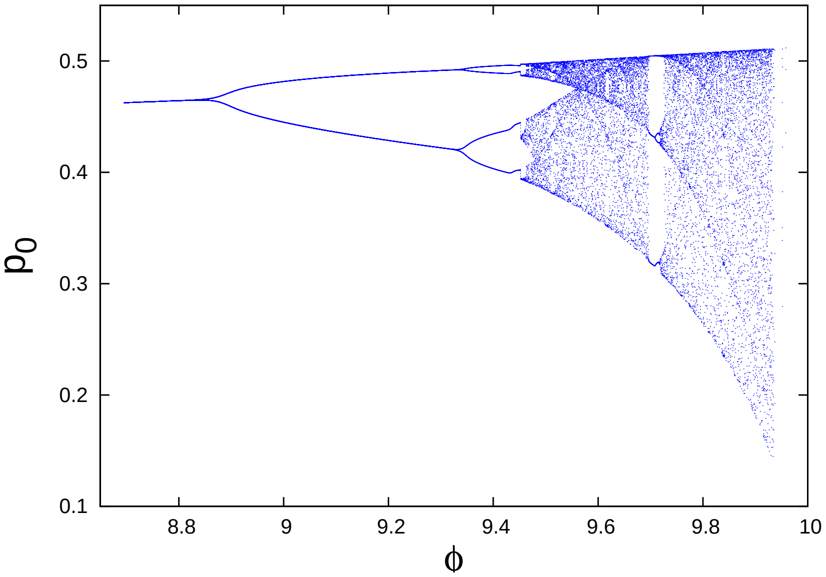


Figure 4.6: Peak values of p_0 plotted as functions of ϕ.

For $8.7 < \phi < 8.9$ the system reaches a nonzero steady state. For larger ϕ, say $\phi = 9.2$, the system now makes an oscillation in which u_0 repeats every two iterations, is called a *period-2 cycle*. At still larger ϕ, say $\phi = 9.4$, the population approaches a cycle that the previous cycle has doubled its period to *period-4*.

Further period-doublings to cycles of period $8, 16, 32, ...$ occur as ϕ increases.

ϕ_1 =8.865	*period 1*
ϕ_2 =9.335	*period 2*
ϕ_3 =9.429	*period 4*
$\vdots$	$\vdots$
ϕ_∞ =9.452..	*period* ∞

These values yield the fraction:

$$\delta = \frac{\phi_2 - \phi_1}{\phi_3 - \phi_2} \simeq 5$$

which is close to the value of 4.669... found by Feigenbaum [46] for the universal constant characterizing the cascade of period-doubling bifurcations leading to chaos.

Ultimately, the successive bifurcations come faster and faster, and ϕ converges to a limiting value ϕ_∞. This value has been found by using the Eq. 4.1 with the approximation at ϕ_∞.

$$\phi_\infty - \phi_n = A\delta^{-n} \tag{4.1}$$

where A is found to be 2.938 from the original equation (Eq. 4.2):

$$\phi_n - \phi_{n-1} = A(\delta^{-(n-1)} - \delta^{-n}) \tag{4.2}$$

Suppose we use our model to generate the subsequent trajectories with the oscillations of the p_0 protein for corresponding ϕ_n values. The results are plotted in Figure 4.7.

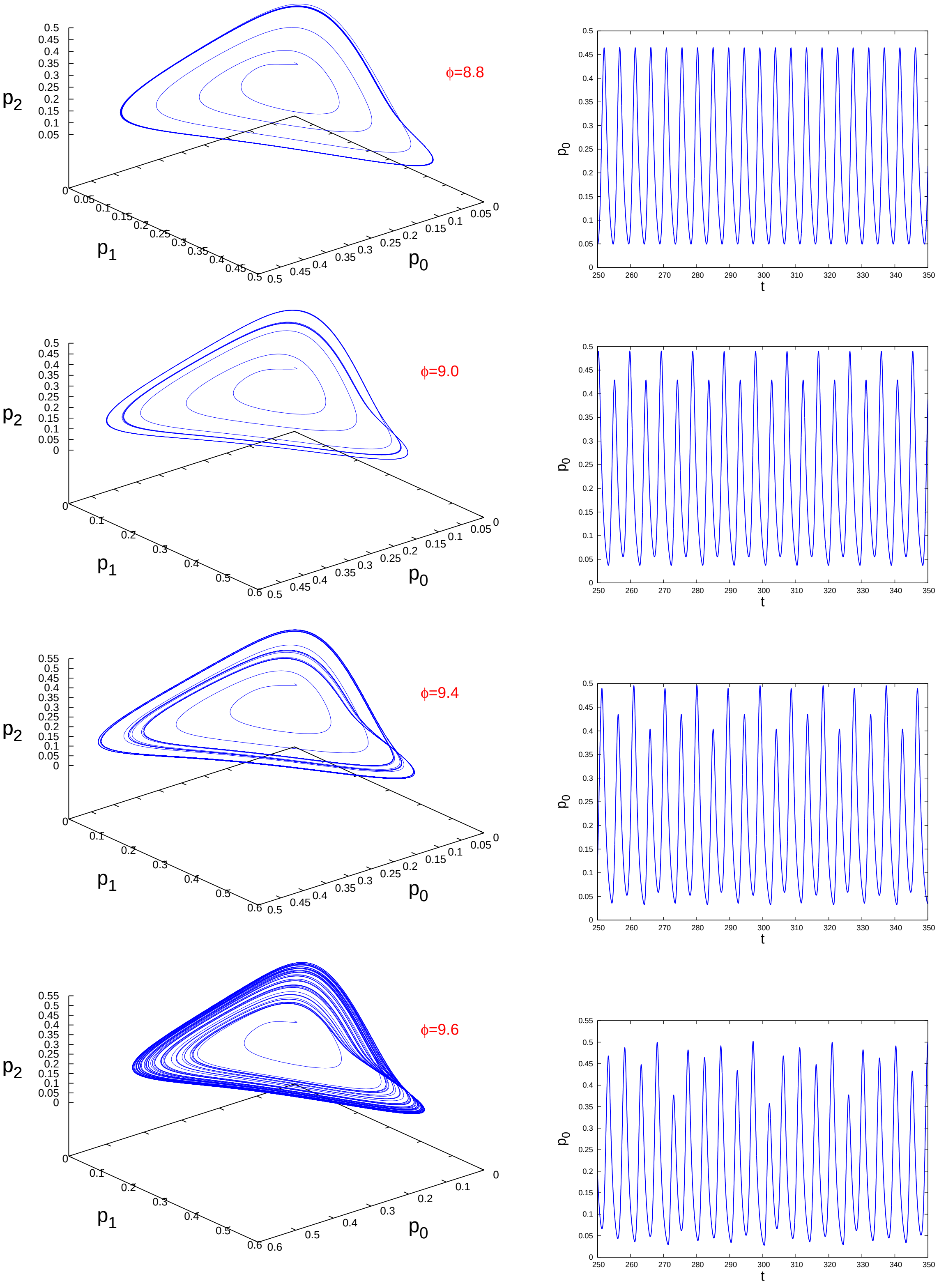


Figure 4.7: Trajectories for four different values of the ϕ parameter. It can be observed that *period 1* oscillation is seen at $\phi = 8.8$, *period 2* oscillation is seen at $\phi = 9.0$, *period 4* oscillation is seen at $\phi = 9.4$, and chaotic oscillation is seen at $\phi = 9.6$.

Furthermore, on the bifurcation plot (Figure 4.6), there is a white window corresponding to $\phi = 9.7$ at the chaotic region. It is a *period 3* window, which arises after the period-doubling route to chaos, then eventually yields to chaos again. The trajectory and the dynamical plots for that window, are shown in Figure 4.8.

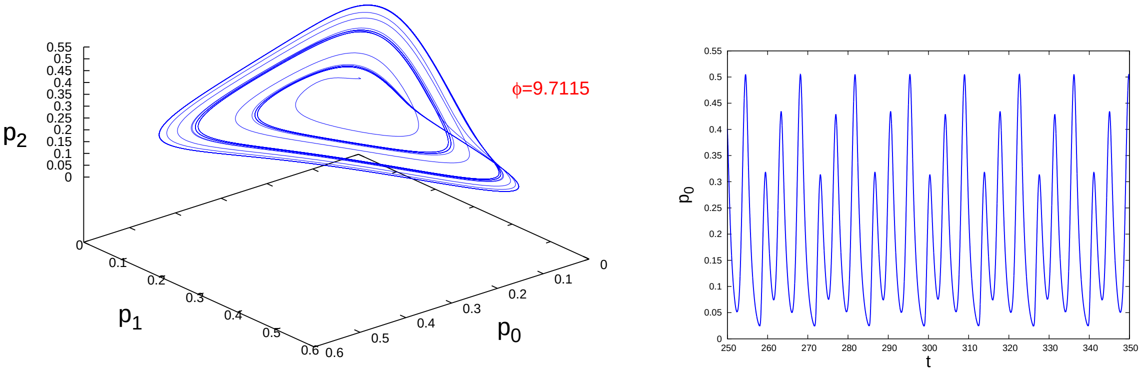


Figure 4.8: The trajectory and the dynamical plots for *period 3* window.

Comparison of Results: By a parameter modification on our model (in Section 2.5), we generated the same bifurcation diagram (Figure 4.6) for the small chaotic network demonstrated in Zhang *et. al.* [1] as shown in Figure 4.9. Notice that, we called the 0^{th} protein as p_0 in Figure 4.6, but they used p_1 notation instead.

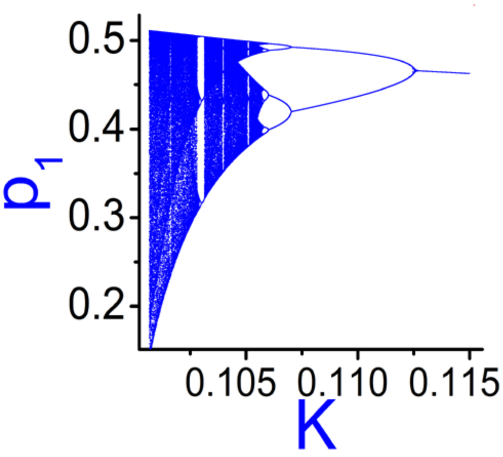


Figure 4.9: Peak values of p_1 plotted as functions of K. Ref.[1, p.3, Fig.1-B]

5. DISCUSSION AND CONCLUSION

The inherent complexity of biological phenomena requires looking at the whole picture, applying various abstraction mechanisms and using computational tools. In this context, the cell can be studied at the systems level by unraveling the regulatory, signaling and metabolic interactions, and their coordinated action. Prediction, control, and understanding arise mainly from modeling these systems using iterated computer simulations.

The main objective of this work was developing a software program that can generate small oscillatory networks, and applying the stochastic optimization to design networks with objective periods. Our results shows that, predicting the structures of a family of networks with desired functionalities through iterative computation, is achievable within at most two days. Although it necessitates parallel computing or longer run times for GRNs with high number of genes, our script could be applied to investigate the dynamics of small networks. As a complementary task, phase space analysis has been performed to infer the operation of a network under parameter changes, and the same bifurcation diagram for the small chaotic network demonstrated in [1] has been generated by a parameter modification on our model.

The research which uses the similar modeling approach [57] is composed of only inhibitory circuits. However, transcription networks often have comparable number of activation and repression controls. Thus, our work includes both inhibitory and activatory regulations as an ideal framework. Also, due to computational limitations, we worked on the networks with 4 and 5 genes; in contrast to the study [57] in which networks consisted of 10 and 20 genes. Therefore, our results are not directly comparable with that study. On the other hand, it has been shown by the statistical analysis that, even considering the effect of activatory links, repressilator motif still plays a dominant role in the dynamics of networks by generating the period $T = 10.5$ as stated in [57].

The most significant aspect of this thesis is the development of a software package that can be used as a computational platform to study dynamical properties of GRNs in a wider context than the topics which have been chosen to apply in this work. It should be remarked that we opted to use an open-source approach for the software development. The main code has been written in the Python programming language. Readily available and convenient modules of Python, such as Numpy, Odespy, Networkx have been used. For the analysis of the results, we have also written several bash scripts in a Linux platform and used other open-source utilities such as Gnuplot. Although issues such as optimization and parallelization have not been considered so far, they can be done in the future for higher performance gains.

As possible developments left for the the future, evaluation of robustness against noise and graph theoretical analysis could be performed on the generated family of networks. Furthermore, inclusion of spatial degrees of freedom into the GRN dynamics would be a very interesting endeavor. In particular, using an inter-cellular communication mechanism such as diffusion, it would be possible to investigate important topics such as synchronization [72] and Turing-like pattern formation [73] in multi-cellular oscillatory systems.

APPENDIX A. Metropolis-Hastings Simulated Annealing

Basically, simulated annealing is a M-H with stationary distribution $exp(\frac{f(x_j^*)}{T})$ for a fixed T.

Main Idea

- Given a parameter $T > 0$ (often called temperature), we draw x_{j+1}^* from $exp(\frac{f(x_j^*)}{T})$
- Function f is the function to maximize.
- As T goes to zero, the values simulated from this distribution become more concentrated around a narrower neighborhood of the local maxima of f.

Pseudocode

```
1. Initialize the algorithm with an arbitrary value $x_0$ and $M$ .
2. Set $j = 1$.
3. Generate $x_j^*$ from symmetric $q(x_{j-1}, x_j^*)$ from $u[0,1]$ .
4. If $u \leq \{exp(\Delta f \frac{x_j^*}{T_j}), 1\}$ then $x_j = x_j^*$, otherwise $x_j = x_{j-1}$ .
5. If $j \leq M$ then $j \rightsquigarrow j+1$, $T_j \rightsquigarrow T_{j+1}$, and go to 3.
```

REFERENCES

1. Zhang, Z., W. Ye, Y. Qian, Z. Zheng, X. Huang, and G. Hu, “Chaotic motifs in gene regulatory networks,” *Plos one*, Vol. 7, no. 7, 2012.

2. Jacob, F., and J. Monod, “On the regulation of gene activity,” *In Cold Spring Harbor Symposia on Quantitative Biology*, Vol. 26, pp. 193–211, 1961.

3. Britten, R. J., and E. H. Davidson, “Gene regulation for higher cells: a theory,” *Science*, Vol. 165, no. 891, pp. 349–357, 1969.

4. Venter, J. C., M. D. Adams, and E. W. Myers, “The sequence of the human genome,” *Science*, Vol. 291, no. 5507, pp. 1304–1351, 2001.

5. Brown, P. O., and D. Botstein, “Exploring the new world of the genome with DNA microarrays,” *Nature genetics*, Vol. 21, pp. 33–37, 1999.

6. Zhu, H., and M. Snyder, “Protein arrays and microarrays,” *Current opinion in chemical biology*, Vol. 5, no. 1, pp. 40–45, 2001.

7. Hood, L., “Systems biology: integrating technology, biology, and computation,” *Mechanisms of ageing and development*, Vol. 124, no. 1, pp. 9–16, 2003.

8. Kitano, H., “Computational systems biology,” *Nature*, Vol. 420, no. 6912, pp. 206–210, 2002.

9. Kitano, H., “Systems biology: a brief overview,” *Science*, Vol. 295, no. 5560, pp. 1662–1664, 2002.

10. Levchenko, A., “Dynamical and integrative cell signaling: challenges for the new biology,” *Biotechnology and bioengineering*, Vol. 84, no. 7, pp. 773–782, 2003.

11. Hasty, J., D. McMillen, F. Isaacs, and J. J. Collins, “Computational studies of gene regulatory networks: in numero molecular biology,” *Nature Reviews Genetics*, Vol. 2, no. 4, pp. 268–279, 2001.

12. Thomas, R., “Regulatory networks seen as asynchronous automata: a logical description,” *Journal of theoretical biology*, Vol. 153, no. 1, pp. 1–23, 1991.

13. Kauffman, S. A., “Metabolic stability and epigenesis in randomly constructed genetic nets,” *Journal of theoretical biology*, Vol. 22, no. 3, pp. 437–467, 1969.

14. Karlebach, G., and R. Shamir, “Modelling and analysis of gene regulatory networks,” *Nature Reviews Molecular Cell Biology*, Vol. 9, no. 10, pp. 770–780, 2008.

15. Li, Z., S. Bianco, Z. Zhang, and C. Tang, “Generic properties of random gene regulatory networks,” *Quantitative Biology*, Vol. 1, no. 4, pp. 253–260, 2013.

16. Remy, E., P. Ruet, L. Mendoza, D. Thieffry, , and C. Chaouiya, “From logical regulatory graphs to standard petri nets: Dynamical roles and functionality of feedback circuits,” *Transactions on Computational Systems Biology*.

17. Steggles, L. J., R. Banks, O. Shaw, and A. Wipat, “Qualitatively modelling and analysing genetic regulatory networks: a Petri net approach,” *Bioinformatics*, Vol. 23, no. 3, pp. 336–343, 2007.

18. Alon, U., *An Introduction to Systems Biology: Design Principles of Biological Circuits*, mathematical and computational biology series, Chapman{&}Hall/CRC, 2007.

19. Bruggeman, F. J., and H. V. Westerhoff, “"The nature of systems biology,” *TRENDS in Microbiology*, Vol. 15, no. 1, pp. 45–50, 2007.

20. Cho, K. H., and O. Wolkenhauer, “Analysis and modelling of signal transduction pathways in systems biology,” *Biochemical Society Transactions*, Vol. 31, no. 6, pp. 1503–1509, 2003.

21. Gardner, T. S., C. R. Cantor, and J. J. Collins, “Construction of a genetic toggle switch in Escherichia coli,” *Nature*, Vol. 403, no. 6767, pp. 339–342, 2000.

22. Elowitz, M. B., and S. Leibler, “A synthetic oscillatory network of transcriptional regulators,” *Nature*, Vol. 403, no. 6767, pp. 335–338, 2000.

23. Sprinzak, D., and M. B. Elowitz, “Reconstruction of genetic circuits,” *Nature*, Vol. 438, no. 7067, pp. 443–448, 2005.

24. Ellis, T., X. Wang, and J. J. Collins, “Diversity-based, model-guided construction of synthetic gene networks with predicted functions,” *Nature biotechnology*, Vol. 27, no. 5, pp. 465–471, 2009.

25. Purnick, P. E., and R. Weiss, “The second wave of synthetic biology: from modules to systems,” *Nature reviews Molecular cell biology*, Vol. 10, no. 6, pp. 410–422, 2009.

26. Locke, J. C., M. M. Southern, L. Kozma-Bognár, V. Hibberd, P. E. Brown, M. S. Turner, and A. J. Millar, “Extension of a genetic network model by iterative experimentation and mathematical analysis,” *Molecular systems biology*, Vol. 1, no. 1, 2005.

27. Fu, P., and S. Panke, *Systems Biology And Synthetic Biology*, Hoboken, New Jersey: John Wiley & Sons, Inc., 2009.

28. Hache, H., *Computational Analysis of Gene Regulatory Networks.* PhD thesis, Humboldt University, Berlin, Germany, 2009.

29. Berge, C., *The Theory of Graphs*, Mineola, NY: Courier Dover Publications, 2001.

30. Thomas, R., “On the relation between the logical structure of systems and their ability to generate multiple steady states or sustained oscillations,” in *Numerical methods in the study of critical phenomena*, pp. 180–193, Springer, 1981.

31. De Jong, H., “Modeling and simulation of genetic regulatory systems: a literature review,” *Journal of computational biology*, Vol. 9, no. 1, pp. 67–103, 2002.

32. Kramer, B. P., A. U. Viretta, M. D.-E. Baba, D. Aubel, W. Weber, and M. Fussenegger, “An engineered epigenetic transgene switch in mammalian cells,” *Nature biotechnology*, Vol. 22, no. 7, pp. 867–870, 2004.

33. Stricker, J., S. Cookson, M. R. Bennett, W. H. Mather, L. S. Tsimring, and J. Hasty, “A fast, robust and tunable synthetic gene oscillator,” *Nature*, Vol. 456, no. 7221, pp. 516–519, 2008.

34. Tigges, M., T. T. Marquez-Lago, J. Stelling, and M. Fussenegger, “A tunable synthetic mammalian oscillator,” *Nature*, Vol. 457, no. 7227, pp. 309–312, 2009.

35. Boyce, W. E., R. C. DiPrima, and C. W. Haines, *Elementary differential equations and boundary value problems*, New York: Wiley, 5 ed., 1992.

36. Tyson, J. J., and H. G. Othmer, "The dynamics of feedback control circuits in biochemical pathways," *Prog. Theor. Biol*, Vol. 5, no. 1, pp. 1–62, 1978.

37. Scott, S. K., *Chemical chaos*, no. 24, Oxford University Press, 1993.

38. Field, R. J., and L. Györgyi, *Chaos in chemistry and biochemistry*, World Scientific, 1993.

39. Olsen, L. F., and H. Degn, "Chaos in biological systems," *Quarterly reviews of biophysics*, Vol. 18, no. 02, pp. 165–225, 1985.

40. Degn, H., A. V. Holden, and L. F. Olsen, *Chaos in biological systems*, Vol. 138, Springer, 1987.

41. Glass, L., and M. C. Mackey, *From clocks to chaos: the rhythms of life*, Princeton University Press, 1988.

42. Kauffman, S., C. Peterson, B. Samuelsson, and C. Troein, "Random boolean network models and the yeast transcriptional network," *Proceedings of the National Academy of Sciences*, Vol. 100, no. 25, pp. 14796–14799, 2003.

43. Strogatz, S. H., *Nonlinear dynamics and chaos: with applications to physics, biology and chemistry*, Perseus publishing, 2001.

44. Kaplan, D., and L. Glass, *Understanding nonlinear dynamics*, Vol. 19, Springer, 1995.

45. Sethna, J. P., "Entropy, order parameters, and complexity," *Statistical Mechanics, Laboratory of Atomic and Solid State Physics, Cornell University, Ithaca, NY*, pp. 14853–2501, 2006.

46. Feigenbaum, M. J., "Quantitative universality for a class of nonlinear transformations," *Journal of statistical physics*, Vol. 19, no. 1, pp. 25–52, 1978.

47. Hinze, T., S. Hayat, T. Lenser, N. Matsumaru, and P. Dittrich, "Hill kinetics meets p systems: A case study on gene regulatory networks as computing agents in silico and in vivo," pp. 320–335, 2007.

48. Bisswanger, H., *Enzyme kinetics: principles and methods*, Weinheim: WILEY-VCH Verlag GmbH & Co. KGaA, 2nd ed., 2008.

49. Keener, J. P., and J. Sneyd, *Mathematical physiology*, Vol. 8, Springer, 1998.

50. Wolkenhauer, O., *Systems Biology-Dynamic Pathway Modelling*, 2008.

51. Goldbeter, A., *Biochemical Oscillations and Cellular Rhythms*, Cambridge, UK: Cambridge University Press, 1 ed., 1997.

52. Goodwin, B. C., "Oscillatory behavior in enzymatic control processes," *Advances in enzyme regulation*, Vol. 3, pp. 425–437, 1965.

53. Griffith, J., "Mathematics of cellular control processes i. negative feedback to one gene," *Journal of theoretical biology*, Vol. 20, no. 2, pp. 202–208, 1968.

54. Sneppen, K., and G. Zocchi, *Physics in molecular biology.*, Cambridge, UK: Cambridge University Press, 2005.

55. Sherman, M. S., and B. A. Cohen, "Thermodynamic state ensemble models of cis-regulation.," *PLoS computational biology*, Vol. 8, no. 3, 2012.

56. Fall, C.P. and Marland, E.S., Wagner J.M. and Tyson, J.J., ed., *Computational Cell Biology (Interdisciplinary Applied Mathematics) (v. 20)*, Springer, 2002.

57. Kobayashi, Y., T. Shibata, Y. Kuramoto, and A. S. Mikhailov., “Evolutionary design of oscillatory genetic networks.,” *The European Physical Journal B-Condensed Matter and Complex Systems*, Vol. 76, no. 1, pp. 167–178, 2010.

58. Monod, J., J. Wyman, and J.-P. Changeux, “On the nature of allosteric transitions: a plausible model,” *Journal of molecular biology*, Vol. 12, no. 1, pp. 88–118, 1965.

59. Hill, A. V., “The combinations of haemoglobin with oxygen and with carbon monoxide,” *The Journal of Physiology*, Vol. 40, pp. 4–7, 1910.

60. Kobayashi, Y., T. Shibata, Y. Kuramoto, and A. S. Mikhailov, “Robust network clocks: Design of genetic oscillators as a complex combinatorial optimization problem,” *Physical Review E*, Vol. 83, no. 6, 2011.

61. François, P., and V. Hakim, “Design of genetic networks with specified functions by evolution in silico.,” *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 101, no. 2, pp. 580–585, 2004.

62. Paladugu, S. R., V. Chickarmane, A. Deckard, J. P. Frumkin, M. McCormack, and H. M. Sauro, “In silico evolution of functional modules in biochemical networks,” *IEE Proceedings-Systems Biology*, Vol. 153, no. 4, pp. 223–235, 2006.

63. Rodrigo, G., J. Carrera, and S. F. Elena, “Network design meets in silico evolutionary biology,” *Biochimie*, Vol. 92, no. 7, pp. 746–752, 2010.

64. Cao, H., F. J. Romero-Campero, S. Heeb, M. Camara, and N. Krasnogor, “Evolving cell models for systems and synthetic biology,” *Systems and synthetic biology*, Vol. 4, no. 1, pp. 55–84, 2010.

65. Sîrbu, A., H. J. Ruskin, and M. Crane, “Comparison of evolutionary algorithms in gene regulatory network model inference,” *BMC bioinformatics*, Vol. 11, no. 1, p. 59, 2010.

66. Jansen, T., *Analyzing Evolutionary Algorithms*, Springer-Verlag Berlin Heidelberg: Springer, 1st ed., 2013.

67. Kirkpatrick, S., “Optimization by simulated annealing: Quantitative studies,” *Journal of statistical physics*, Vol. 34, no. 5-6, pp. 975–986, 1984.

68. Bornholdt, S., and K. Sneppen, “Robustness as an evolutionary principle,” *Proceedings of the Royal Society of London. Series B: Biological Sciences*, Vol. 267, no. 1459, pp. 2281–2286, 2000.

69. Roberts, G. O., A. Gelman, W. R. Gilks, *et al.*, “Weak convergence and optimal scaling of random walk metropolis algorithms,” *The annals of applied probability*, Vol. 7, no. 1, pp. 110–120, 1997.

70. Rossum, Guido van, e. a., “Python language reference, python software foundation.” `http://docs.python.org/2.7/index.html`.

71. Hagberg, A. A., D. A. Schult, and P. J. Swart, “Exploring network structure, dynamics, and function using NetworkX,” in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, (Pasadena, CA USA), pp. 11–15, Aug. 2008.

72. Garcia-Ojalvo, J., M. B. Elowitz, and S. H. Strogatz, “Modeling a synthetic multicellular clock: repressilators coupled by quorum sensing,” *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 101, no. 30, pp. 10955–10960, 2004.

73. Hsia, J., W. J. Holtz, D. C. Huang, M. Arcak, and M. M. Maharbiz, “A feedback quenched oscillator produces turing patterning with one diffuser,” *PLoS computational biology*, Vol. 8, no. 1, p. e1002331, 2012.