A MATHEMATICAL MODELING AND APPROXIMATION OF GENE EXPRESSION PATTERNS BY LINEAR AND QUADRATIC REGULATORY RELATIONS AND ANALYSIS OF GENE NETWORKS

FATMA BÍLGE YILMAZ

AUGUST 2004

A MATHEMATICAL MODELING AND APPROXIMATION OF GENE EXPRESSION PATTERNS BY LINEAR AND QUADRATIC REGULATORY RELATIONS AND ANALYSIS OF GENE NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

FATMA BÍLGE YILMAZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF SCIENTIFIC COMPUTING

AUGUST 2004

Approval of the Graduate School of Applied Mathematics

Prof. Dr. Aydın AYTUNA
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Bülent KARASÖZEN
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Gerhard-Wilhelm WEBER
Supervisor

Examining Committee Members

Assoc. Prof. Dr. Tanıl Ergenç

Prof. Dr. Bülent Karasözen

Assist. Prof. Dr. Erkan Mumcuoğlu

Dr. Hakan Öktem

Prof. Dr. Gerhard-Wilhelm Weber

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Fatma Bilge Yılmaz

Signature:

ABSTRACT

A MATHEMATICAL MODELING AND APPROXIMATION OF GENE EXPRESSION PATTERNS BY LINEAR AND QUADRATIC REGULATORY RELATIONS AND ANALYSIS OF GENE NETWORKS

Fatma Bilge Yılmaz
M.Sc., Department of Scientific Computing
Supervisor: Prof. Dr. Gerhard-Wilhelm Weber

August 2004, 86 pages

This thesis mainly concerns modeling, approximation and inference of gene regulatory dynamics on the basis of gene expression patterns. The dynamical behavior of gene expressions is represented by a system of ordinary differential equations. We introduce a gene-interaction matrix with some nonlinear entries, in particular, quadratic polynomials of the expression levels to keep the system solvable. The model parameters are determined by using optimization. Then, we provide the time-discrete approximation of our time-continuous model. We analyze the approximating model under the aspect of stability. Finally, from the considered models we derive gene regulatory networks, discuss their qualitative features of the networks and provide a basis for analyzing networks with nonlinear connections.

Keywords: Gene Expression, Gene Regulation, Mathematical Modeling, Gene Network, Inference, Optimization, Dynamical Systems.

ÖZ

GENE MOTİFLERİNİN DOĞRUSAL VE İKİNCİ DERECE DÜZENLEYİCİ İLİŞKİLERLE MATEMATİKSEL MODELLENMESİ VE MOTİFLERE YAKLAŞILMASI VE GENLERİN OLUŞTURDUĞU AĞ YAPILARININ ANALİZİ

Fatma Bilge Yılmaz
Yüksek Lisans, Bilimsel Hesaplama Bölümü
Tez Yöneticisi: Prof. Dr. Gerhard-Wilhelm Weber

Ağustos 2004, 86 sayfa

Bu tez esas olarak, gen motiflerini baz alarak, gen düzenleyici dinamiğinin modellenmesi, sistemi belirleyen denklemlere yaklaşılması ve onun hakkında çıkarımda bulunulması hakkındadır. Gen motiflerinin dinamik davranışları olağan differensiyel denklemlerle gösterilir. Sistemi çözülebilir tutmak için, bazı doğrusal olmayan alanlara sahip gen etkileşim matrisleri kullanılır, bu alanlar özel olarak gen seviyelerine bağımlı ikinci derece polinomlardır. Model parametrelerini optimizasyon kullanarak buluruz. Daha sonra, kesintisiz zamanda evrilen modelimiz için kesintili zamanda evrilen bir yaklaşım sağlarız. Kesintili zamanda evrilen modelimizi kararlılık yönünü göz önüne alarak inceleriz. Son olarak, önceden geliştirilmiş modellerle gen düzenleyici ağlarını elde eder, bu ağların nitel özelliklerini tartışır ve doğrusal olmayan bağlantıları olan gen ağlarının incelenmesine esas bilgiler veririz.

Anahtar Kelimeler: Gen Motifleri, Gen Düzenlemesi, Gen Ağ Yapıları, Matematiksel Modelleme, Çıkarım, Optimizasyon, Dinamik Sistemler.

To my family

ACKNOWLEDGMENT

I would like to thank all those people who have helped in the preparation of this study. I am grateful to Prof. Dr. Bülent Karasözen who introduced me nice friends in the world of mathematics. To Assoc. Prof. Dr. Tanıl Ergenç I express my gratitude for this thesis and valuable remarks on my research.

I would like to thank Prof. Dr. Gerhard-Wilhelm Weber who has always encouraged me for my study. I am pleased that I have a very kind, friendly advisor and also a good friend like him. I would like to thank him for all his efforts, energy and support. I am very glad that I meet Prof. Dr. Gerhard-Wilhelm Weber and Prof. Dr. Marat Akhmet who always make me laugh with their nice jokes. Also, I thank to nice friends of Prof. Dr. Gerhard-Wilhelm Weber, especially to Dr. Stefan Pickl, who try to help and coorporate with us.

Special thanks to Dr. Hakan Öktem for all his efforts in every phase of my study. I would like to thank him for his help and patience when teaching and listening. Thanks to Dr. Hakan Öktem because he always has some answers to all my questions. I am very glad that I have such an advisor. Also, I would like to thank him for his precious friendship.

I need to point out that my company, Turkcell has always supported me financially during my study and I would like to thank all my friends in my office for their patience and support. I would like to thank Recep Acar who supported me during the study. Special thanks to Hüseyin Yılmaz Ulus for his existence which always gives me hope.

I would like to thank my family for their endless support, hope, care, help and love. I would like to thank my brother Mehmet Alper Yılmaz for his sympathy and to my mother Şadan Yılmaz and my father Murat Yılmaz for their efforts. I would like to dedicate this study to my family whom I am proud of.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background to the Study

The genome of an organism plays a crucial role in the regulation of cellular processes like *adaptation*, *differentiation*, *development* and *maintenance* [7, 53, 79]. The main role in these processes is the information flow from the genome to the proteins, i.e., *protein synthesis*. Proteins synthesized by transcription of genes control almost every cellular function, but here we underline two important functions related to the regulation mechanisms.

According to the current metabolic state, a cell activates or deactivates some particular biochemical reactions [79]. For example, in the absence of dietary glucose, glucose availability should be maintained by activating related biochemical reactions. These reactions are catalyzed by enzymes most of which are proteins. Enzymes stabilize the transition states of the reactions by increasing the rate of the reactions. Biochemical reactions without enzymes usually take a very long time compared to the catalyzed reactions.

Here, the most significant question is how cells adjust the expression of genes in response to different environmental conditions. The answer is *gene regulation*. Gene regulation is introduced by *Jacob, Lwoff and Monod* in [52], where they discovered that proteins bind to regulatory regions of other genes and function as *transcription factors*. This discovery constitutes the second important function related to genes. For example, *phosphofructokinase* is an enzyme which controls the flux of metabolites through the glycolytic pathway.

These roles imply that proteins generate a regulation network where they not only regulate the cellular processes, but the availability of each other as well [52, 79]. Although we define the regulation network in terms of proteins, we underline that genes constitute a basis for all regulatory factors in biological processes since proteins are encoded by genes. Thus, a regulation network should be constructed by means of gene interactions.

Nowadays, *biochips* are mentioned and enthusiastically praised in television, magazines and newspapers. There is a big public attention and fashion which almost expect medical wonders from this technology, regarded as a breakthrough combination of biology and computer engineering. However, what is mathematically behind such high expectations? The emergence of new high-throughput functional genomics technologies is a crucial driving factor in our ability to understand gene interactions [7, 44, 53]. Microarray technologies provide us with gene expression profiles which are reflections of the dynamical behavior of biological systems and the underlying regulatory network of interacting genes [44, 64].

Achieving an understanding of gene regulation requires more than merely collecting the gene expression profiles. The dynamics of the underlying gene interaction network, chemical structure of the enzymes, allosteric regulations, enzymatic reaction rates, environmental factors that affect the regulation mechanisms should be studied to gain such an understanding. Mathematical modeling of such a complex system is a key issue in bioinformatics that deals with the time-series gene expression patterns to infer the underlying gene regulatory network.

1.2 Current Mathematical Research Studies

The idea of mathematical description of biological organisms was first introduced by *Turing* in 1952 [20, 75]. Turing stated that the change of state of a cell is the sum of all forces acting on that cell, which is a foundation for regulation mechanisms. Until 1952, different model classes reflecting the dynamics of genetic regulatory networks have been studied, including Boolean networks, Bayesian networks, stochastic models and differential equations.

The most widespread formalism is the latter modeling approach, i.e., differential equations, because of the qualitative dynamics of regulatory interactions. Regulation mechanisms are composed of interconnected regulatory pathways, and most of the factors which affect the dynamical behavior have not been determined yet [79].

This mystery distinguishes the modeling approach from the others which either make strong assumptions on the structure and dynamics of regulatory interactions or totally disregard the dynamical behavior of the genetic regulatory systems [44]. Another attractive feature of using differential equations is that they evolve in continuous time which is appropriate for the nature of the regulation networks [44, 64]. Response to a constant environmental data in a stable level of gene expressions and differential equations can determine the steady states analytically [49, 64]. This is another critical advance of using differential equations in modeling.

Mathematically modeling such a system with differential equations combined with the theories of inverse problems and statistical learning constitutes our wish to model and anticipate the gene interactions.

Although we underline the advantages of using differential equations to model the dynamical behavior of gene expression patterns, two critical disadvantages of the model come into existence, the computation time [64] and accuracy of the system [71]. The interconnected structure and global view of the biological system is protected by using differential equations which makes the computational drawbacks less important. But some improvements can effectively reduce the computation time. Since gene regulatory matrices are essentially sparse [37], the matrix can be partitioned into appropriate subsystems and this partitioning enables parallel processing. Accuracy deficiency of the system results from the insufficient experimental data. When sufficient source of data is available, differential systems can be solved numerically to any desired precision by adjusting the prediction error between real and approximated data [71].

1.3 Brief Description of the Study

The metabolic state of a cell is usually regulated by gene and enzyme regulation mechanisms in the cell which ensures that the current metabolic state can be used to predict the next metabolic states. Since only a limited portion of the gene interactions are known, we do not have sufficient information to build a first principle model. Our approach, also being preferred by most of the functional genomic modeling researchers, is selecting a model class and inferring the parameters from empirical data. Here, selecting the appropriate model class is the most crucial step. Firstly, estimation of parameters characterizing the dynamics of the system should have a unique solution. Secondly, the model should be able to approach the dynamics of the system as accurate as possible. For example, a linear differential system is not sufficient to seize the nonlinearities in the behavior of the system. On the other hand, the exact model will be included within the class of ”any possible differential equations” but it cannot be distinguished from the infinitely many other solutions fitting to a particular data.

1.4 Purpose of the Study

The contribution of this work consists in continuously approximating the behavior of time-series gene expression patterns by a system of ordinary differential equations based on the approach described in [25]. We use a smooth model given by differential equations where the dynamics of the gene interactions, constituting the right-hand side of that system of ordinary differential equations, is represented by quadratic polynomials. *The inverse problem*, estimation of the polynomial coefficients, is solved by discrete approximation which is *a least squares optimization problem.* The solution of the inverse problem gives us polynomials for each interaction. Finally, we construct *a dynamic regulatory gene network* by means of the polynomials and give a brief discussion about the analysis of gene interactions by *dynamic shortest path algorithms.*

This thesis concerns an efficient approximation to the regulatory dynamics of cells in a particular stationary state. The multistationary nature of genomic regulation was discussed in several articles [71]. Some Boolean and piecewise linear algorithms were proposed for seizing the transitions from one stationary steady state to another. However, the inherent nonlinearities like the effect of enzyme concentrations on reactions has a potential risk of concluding on faulty state transitions with piecewise linear approach, while Boolean models do not provide quantitative solutions. Thus, this thesis aims at providing an efficient approximation to the behavior of cells at a particular stationary steady state. Involving transitions between the metabolic states can then be considered as a further research subject.

1.5 Significance of the Study

There are three crucial properties related to our model. Firstly, we provide a continuous approximation to the experimental data to reflect the natural biological regulatory systems. Secondly, we model gene interactions by polynomials which both guarantee to have a unique solution for the inverse problem and handle a much larger range of dynamics with respect to a linear approximation. The idea comes from the fact that biological systems are quite sensitive. Finally, our model enables inference of a wide range of biological pathways since there is no assumption on the structure and the dynamics of the pathways under consideration.

Unfortunately, the available experimental data is insufficient to make strong conclusions about biochemical reactions in nature. We underline that our model approximates the experimental data and provides us with valuable results for these reactions, such as strong clues about the biological roles of genes, as well as the dependencies between genes.

Given sufficient information, our methodology improved by involving transitions between the metabolic states, the response of a cell to changes in environmental conditions, differentiation and evolutionary factors can be determined. For example, the change in the expression levels of genes in two cells can be analyzed and the results can be used for prediction of healthy and cancer cells, in better medical treatment and drug design. Another outcome is identifying environmental effects and cell capability of adaptation. It must be underlined that with sufficient information about the responses of cells to external and internal stimulus, the utilities of our model will exponentially grow.

CHAPTER 2

BACKGROUND

2.1 Genetic Material

Organisms contain a genetic material which controls both phenotypic and genotypic traits (for details see [45]). There are four major characteristics of this genetic material stated: Replication, variation of mutation, storage and expression of information.

The main characteristic related to genotype is replication which is a source of *heredity.* Genetic material is used as a template for the synthesis of a new strand, i.e., information encoded in genetic material is fully transferred to the offspring by replication.

Expression of information is a foundation for phenotypic processes. A phenotypic process requires some proteins to be synthesized. Information needed for each protein is encoded in some expressive part of the genetic material, called *gene.* Genes which are capable of information storage, serve as a template for the synthesis of related proteins, regulate transcription and make the phenotypic processes possible. In addition to the role in phenotypic processes, proteins have various functions in a cell cycle. However, in this study, we concentrate on the gene regulation mechanism in phenotypic processes.

In the following sections, we will explain how genes are linked to organism phenotype by protein synthesis and clarify the regulating mechanisms in protein synthesis.

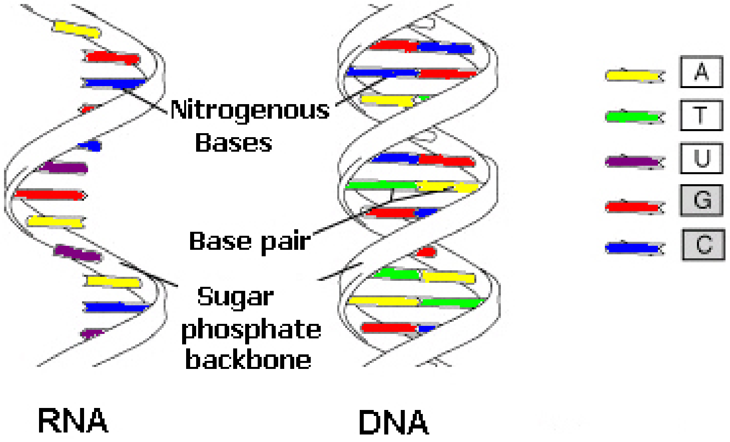


Figure 2.1: DNA and RNA (a modified figure taken from the Genome Research Institute, Genetic Illustrations).

2.2 Central Dogma of Biology

2.2.1 Nucleic Acids

Except some viruses and prions as described in [47], *DNA (Deoxyribose nucleic acid)* is the chemical component that constitutes genetic information for phenotypic traits [6]. Basic unit of DNA is a *nucleotide*, which is composed of a phosphoric acid, a sugar and a nitrogenous base. As shown in Figure 2.1, DNA exists in a double helix where corresponding nucleotides in two helices are linked together with weak hydrogen bonds between base pairs adenine (A)- thymine (T) and guanine (G)- cytosine (C). This special structure was first described by *Watson and Crick* [80], where they introduced the basis for the concept of *complementarity*, i.e., the chemical affinity provided by the hydrogen bonds between the specific base pairs. The complementarity is universal but the source of diversity comes from the variation in the sequence of the nucleotides.

We explained in Chapter 1, that the success of a phenotypic process depends on the existence of some proteins. The genes are the initial chemical components and proteins are the end products in information flow for protein synthesis. *Guyton and Hall* [32] explain that there must be another chemical component in the cell to carry the information from DNA to *ribosome*, the organelle that is responsible of protein synthesis. This material is *RNA (Ribo nucleic acid)*.

RNA is another nucleic acid that is generated, constituted and controlled by DNA and has a similar structure like DNA (see Figure 2.1). The component serves as an intermediate in transferring genetic code from DNA to proteins by means of this structural similarity. RNA is a single-strand chain of nucleotides that contains base urasile (U) instead of T. There are three major types of RNA: *mRNA (messenger RNA)*, whose nucleotide sequence transfers the genetic codes to proteins; *tRNA (transfer RNA)*, which is capable of delivering amino acids to the ribosomes; *rRNA (ribosomal RNA)*, constituting two-third of the ribosomal mass. As statements for the main responsibilities of the RNAs indicate, protein synthesis is a common activity for all types of RNA.

Two intermediate processes come into existence in protein synthesis (see Figure 2.2 and 2.3). The first process is *transcription* of information encoded in one gene into RNA and the second step is *translation* of the information from RNA to a protein. These two processes serve as a foundation for *the central dogma of biology*: Genes encode for proteins via RNA.

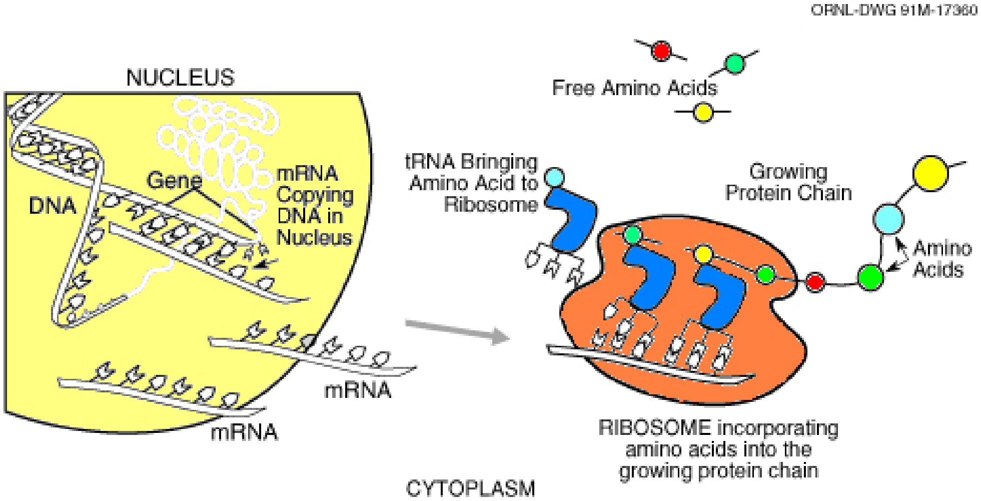

ORNL-DWG 91M-17360
NUCLEUS
DNA
Gene
mRNA Copying DNA in Nucleus
mRNA
mRNA
Free Amino Acids
tRNA Bringing Amino Acid to Ribosome
Growing Protein Chain
Amino Acids
RIBOSOME incorporating amino acids into the growing protein chain
CYTOPLASM

Figure 2.2: Central Dogma of Biology (a modified figure taken from U.S. Department of Energy Human Genome Program).

2.2.2 Transcription

The first step in protein synthesis is transcription. Information necessary for a protein is encoded in one of the DNA strands which we call *antisense (noncoding)* strand. Transcription is initialized by binding of *RNA Polymerase* enzyme to base sequences called *promotor* in antisense strand. RNA Polymerase breaks the hydrogen bonds between nucleotides and opens some portion of the double-stranded DNA. Suitable nucleotides which are available in the environment, are linked to the nucleotides in noncoding DNA strand with hydrogen bonds. Hydrogen bonds exist between bases A-U and G-C as shown in Figure 2.3. Linked nucleotides are activated and added to current incomplete RNA strand, called *pre-mRNA.* This process continues until the stop condition is reached.

The genetic code encoded in the antisense strand is completely transcribed onto its complementary pre-mRNA strand by transcription. After transcription, some processes like adding *poly-A tail* and *splicing* are applied to pre-mRNA (for details see [45]). The resulting sequence is called mRNA. Adding a sequence of A nucleotides is very critical in DNA experiments and regulated by *polyadenylase* enzyme. In the splicing process, some portions of pre-mRNA, called *introns*, are dropped from the sequence since they are useless in protein synthesis. Although adding poly-A tail and splicing seem to cause a lack of information, mRNA corresponds to the *pseudo genes* in DNA strand and it is almost a mirror image of the portion of the DNA strand providing important clues about the related gene.

2.2.3 Translation

After construction of mRNA, mRNA moves to ribosomes to start translation which is the second step in protein synthesis. The smallest meaningful sequence in mRNA is called *codon* and consists of three consecutive nucleotides. For example, to synthesize the amino acid *Gly*, necessary codon consists GGC sequence as shown in Figure 2.3.

Translation can be summarized as the synthesis of an amino acid chain where information necessary for each amino acid is stored in corresponding codon of mRNA. Amino acids are located in the cytoplasm of the cell, but translation takes place on ribosomes. Transfer of the amino acids from cytoplasm to the ribosomes is a responsibility of tRNA. Each tRNA consists of three nucleotide sequences corresponding to an amino acid. Necessary amino acids combine with particular tRNAs and are transferred to ribosomes. Ribosomes are the organelles, where *anticodon* in the tRNA links to the corresponding codon in mRNA to put the amino acid into a suitable place in the amino acid sequence. For example, to synthesize the amino acid Gly, a tRNA that contains CCG sequence is combined with the amino acid and linked to the codon GGC. This is shown in Figure 2.3.

The amino acid is then combined with other amino acids by peptide bonds. After the amino acid is located, mRNA goes forward for the next codon. In the figure, the next codon is UGU and it is used to synthesize the amino acid *Cys*. The corresponding tRNA containing the anticodon ACA is combined with Cys and transfer it to the ribosome for the current incomplete amino acid chain. Translation ends when the last amino acid is bonded to the current amino acid chain.

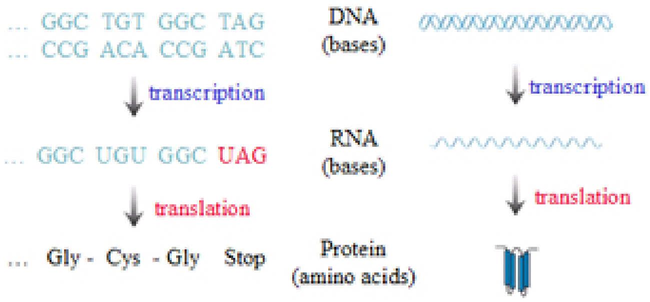


Figure 2.3: Protein synthesis.

In the previous section, we described that the proteins regulate transcription of each other. Thus, a brief summary of enzymes and enzyme kinetics will give an insight into the rate of biological processes, consequently clues about the regulation mechanisms related to enzymes. The following section summarizes enzymes and enzyme kinetics (for details see [79]).

2.3 Enzymes

2.3.1 Enzymes as Catalysts

Enzymes increase the rates of biochemical reactions by lowering the free energy barrier that separates the reactants and products and stabilize the *transition state*, the point of highest free energy of the catalyzed reaction. So, they are usually called *biological catalysts* of livings. Despite of the functional similarity between an enzyme and an ordinary catalyst, enzymes have some distinctive properties compared to ordinary chemical catalysts, such as higher reaction rates, greater reaction specificity and capacity of regulation [79].

2.3.2 Enzyme Kinetics

At constant temperature, the rate of an elementary reaction is proportional to the frequency with which the reactants come together, thus the rate increases when the concentrations of the reactants increase, and vice versa. Environmental changes affect the rate of a chemical reaction. For example, as temperature increases, the energy of each reactant, i.e., the probability of collisions between reactants, increases which results in an increase in the reaction rate. Another example for environmental changes is adding an *inhibitor* to the environment, which reduces the available catalysts in the environment and decreases the rate of the reactions.

Enzymatic reactions in reality pass through several reactions, which are usually indeterminate, so the analysis of these reactions is more complex than elementary reactions. Under *steady state assumption*, i.e., the rate of synthesis and consumption are equal, the rate determining reaction in an enzymatic reaction chain is the reaction with the highest transition state. As ordinary chemical reactions, environmental changes are reflected in the rate of the enzymatic reactions, but the main difference in studying kinetics of an ordinary catalyst and an enzyme is that the transcription regulations are directly affected by the rate of enzymatic reactions. This constitutes the main challenge in studying enzyme kinetics. Another challenge is that the mechanisms which regulate the pathway activity under different physiological conditions, have not been entirely discovered yet.

The study of enzymatic reaction rates combined with information about the chemical structure of the enzymes and related metabolic pathways, gives us important clues about not only the biological function of the enzymes, but also the reasons for changes in transcription rates and, consequently, regulation mechanisms for transcription as well.

2.3.3 Metabolism

Metabolism is the overall process through which organisms acquire and use their free energy to carry out various cellular functions. The principles which govern metabolism are the same in all organisms according to common evolutionary origin and constraints of thermodynamics laws, as explained in [79]. Except some small variations in source of free energy, many of the metabolic reactions are essentially identical in all organisms.

A metabolism can also be seen as an integrated and regulated metabolic pathways each contributing to the cellular activities. *Metabolic pathways* are series of biochemical reactions that produce and consume specific products. A simple classification of the pathways can be made depending on whether they consume or produce energy. The metabolic pathways which concern with the synthesis of cellular components are called *anabolic pathways,* while *catabolic pathways* are involved in the breakdown of cellular constituents.

Multistep metabolic pathways are usually irreversible and have independent control mechanisms for anabolic and catabolic pathways. This comes from the fact that an *exergonic (irreversible)* reaction early in a multistep metabolic pathway makes the entire pathway irreversible and requires different anabolic pathway corresponding to a catabolic pathway, and vice versa. The irreversibility implies that the network of metabolic pathways is strongly connected.

2.4 Regulation of Transcription

In order to respond environmental changes, growth and differentiation, organisms regulate the catalytic activities of enzymes by controlling the availability and activity of enzymes.

Enzyme availability is regulated by the rate of synthesis and degradation of an enzyme. Allosteric mechanisms, the structural alterations which influence the enzyme's binding affinity, alter the enzymatic activity. *Donald et al.* [79] and *Guyton et al.* [32] explain the gene and enzyme regulation mechanisms in detail. In the following sections, we give a brief summary of the regulation mechanisms.

The regulation mechanisms not only include gene and enzyme regulations, but external stimulus, e.g., hormones, as well. For example, *insulin* and *glucagon* are the two major hormones that regulate glycolysis pathway [51]. Glucagon acts to maintain glucose availability in the absence of dietary glucose where insulin is an anabolic hormone that acts in the opposite way. Although regulation networks consists of very complex and integrated regulation mechanisms, we underline that gene regulation is a very important factor in regulation mechanisms since gene transcription regulates transcription factors, which in turn regulate gene transcription.

2.4.1 Enzyme Regulation

The first regulator mechanism in a cell is enzyme regulation. Main characteristics related to enzyme regulation are feedback loops in biochemical reactions and self-regulating substances. In biochemical synthesis, usually the synthesized substance can deactivate the first necessary enzyme for the reactions. This inhibition makes a decrease in concentrations of the intermediate substances by deactivating transcription and regulates the synthesis. In addition to this, some of the biochemical substances also have an ability to inhibit the enzymes which syntheses themselves.

2.4.2 Gene Regulation

Expression of a gene depends on the activation of *operon* sequences. Activation and degradation of operon sequences are regulated by the *promotor* region. There are two main regions in promoter: *repressor* and *activator operators*. A repressor operator is critical in inhibition of protein synthesis because, if *a repressor protein* binds to this region, protein synthesis will be deactivated. There are *activator proteins* in the cell to break this bond and activate the transcription. An activator operator in promoter helps RNA polymerase to bind to promoter region, activates the operon and makes transcription possible.

If the synthesized material quantity reaches a critical value, the operon which is responsible for the synthesis of this material is inhibited by a *negative feedback mechanism*, either breaking the bonds between activator protein and operator, or binding repressor protein to repressor operator. In any case, operon is deactivated and protein synthesis is controlled.

Figure 2.4 is shown in [79] and includes genes *X, Y* and *Z* encoding for three proteins, *$\beta -$ galactosidase*, *galactoside permease and thiogalactosidase transacetylase* for lactose metabolism in *E.coli* and the regions that regulate the expression of these genes. The regulatory gene is not part of the lactose operon but encodes a repressor protein that inhibits transcription of the lac operon. Promotor region is named as control sides, where activator and repressor regions, P and O, in the promotor are shown separately in the figure.

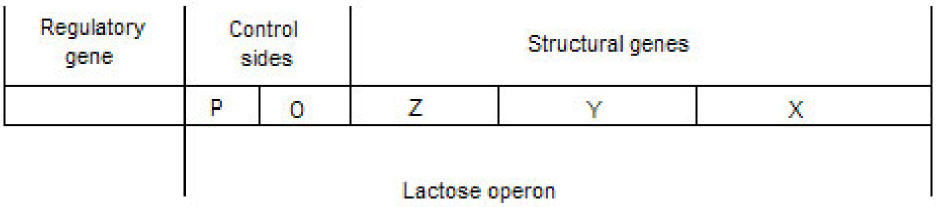


Figure 2.4: Lac operon in E.coli [79].

Synthesis and regulation mechanisms in complex systems lead to hierarchical organization, differentiation and increased functionality [38, 39, 74]. In prokaryotic genomes, genes with related functions are usually located in the same operon, as shown in the figure and transcribed together, while in eukaryotes most of the protein coding genes are transcribed individually. We will analyze how the complexity of organisms, i.e., the complexity of synthesis and regulation mechanisms, affects mathematical modeling approach in the coming sections.

If we have an information about the expression levels of genes, this information provides us an understanding about the gene interactions, i.e., gene regulation mechanisms. DNA arrays technology enables *gene expression profiling*, measuring the level of the mRNA gene products of a living cell simultaneously [7]. The next section summarizes this technology and discusses the importance of the gene expression profiling.

2.5 DNA Microarray Experiments

2.5.1 Purpose of the Experiments

DNA microarrays are designed to observe the gene expression of an experimental cell compared to a reference cell. For example, adding a suitable nutrient to an experimental cell makes a metabolic shift in the cell, by inhibiting or stimulating expressions levels of the genes, while metabolic state of the reference cell remains the same. A small example is shown on the left-hand side in Figure 2.5. After adding a nutrient, transcription rate of Gene A is lowered, while the expression level of Gene B becomes increased and Gene C does not give a response to the environmental change.

Another example, shown on the right-hand side of Figure 2.5, simulates the difference in the expression levels in normal and cancer cells. Since cancer cells lose the original phenotypic characteristics; in this figure, Gene C has some mutations, the expression profiles differ in two cells. In both cases, the change in expression levels of genes in experimental and reference cells can be observed by DNA microarrays technology.

2.5.2 Design and Implementation

There are several steps in design and implementation of a DNA microarray experiment. These steps are shown in Figure 2.6, and they can be mainly listed as array preparation, hybridization and image processing.

Array preparation is the first step in DNA microarray experiments. Suitable cDNA sequences which are obtained from mRNAs of experimental and reference cells, are selected and deposited on distinct glass arrays. Before printing, the glass arrays are treated to enhance binding of the cDNA to the glass surface by air drying and UV-crosslinking procedure, which increases fixation of the cDNA probes to the glass [53]. The last step in array preparation is heating the arrays to separate strands of cDNA.

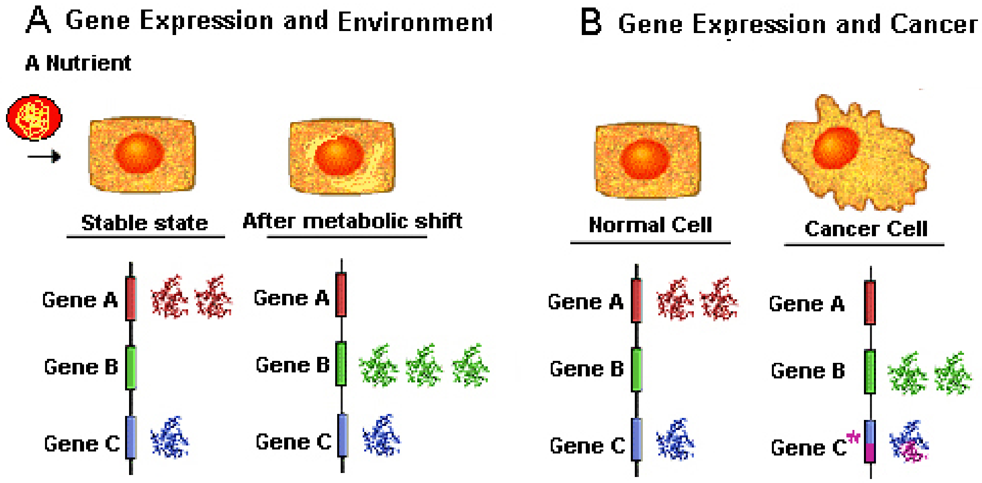


Figure 2.5: Gene expression profiles change according to the environment (A) and in a normal and cancer cells (B) (a modified figures taken from the Cancer Genome Anatomy Project, Conceptual Tour).

The second step is the critical step in the experiments. cDNAs from the experimental and reference samples are labeled with different fluorescent dyes, mixed, and hybridized to probes on the array. *Hybridization* refers to the binding of two cDNA strands by base pairing. After a sufficient time for hybridization and a series of washes to eliminate all unbounded target cDNAs and solution, each probe shows the expression level changes of a gene in experimental cell.

The last step of the experiments includes *image processing techniques.* The array is scanned for image processing analysis which is required to extract the numerical data. This process involves estimating the location of the spot on the array and, then, measuring the spot intensity as well as the background intensity based on the area outside the spot.

DNA microarray experiments provide us time-series experimental data for ex-

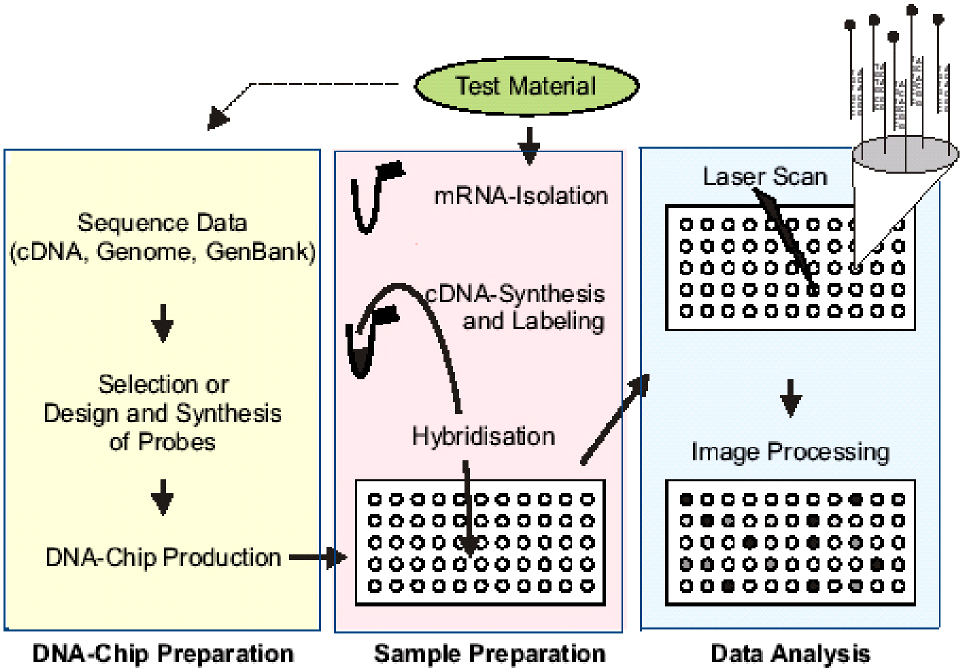


Figure 2.6: Design and implementation of DNA microarray experiments [25].

pression levels of genes, which show the relation between genotype and phenotype and, consequently, helps understanding biological processes [7, 53]. Systems ranging from gene regulation, to development, evolution and diseases, to differentiation as well as annotation of gene functions, identifying the effects of environment and life style, giving an insight into nutritional intake and therapeutic drugs, can be understood by DNA microarray experiments [7].

Our present study concerns the analysis of the gene expression levels and inference for the underlying gene regulatory dynamics. So, our study may serve as a new explanation of this promising modern technology by means of mathematical modeling, dynamical systems, algorithms and networks.

CHAPTER 3

REVIEW OF LITERATURE

3.1 Challenges in Modeling

Given a finite number of gene expression patterns, identifying the underlying dynamics of gene regulatory interactions contains many challenges.

The first challenge is insufficient information about the underlying biochemical reactions, allosteric regulations, chemical structure of the biochemical substances and regulatory dynamics. Although there are some pathways which are relatively better known, such as glycolysis pathway studied in [15, 62], the current information is not sufficient to solve the mystery in the regulatory relations and gain a general perspective of the dynamics.

With current technology, microarray experiments give us valuable information about gene expression patterns and deliver an insight into regulatory relations. The success of a model depends on the quality of the training data. Repetition of the experiments and a large number of experimental data with smaller sampling time will play an important role in eliminating noisy data and modeling the continuous biological processes. The current experimental data are insufficient to describe the natural processes, which constitute the second important challenge in modeling.

Specification and organisation in the organisms is another challenge to be considered. With current technologies in hand, we are able to view gene expression profiles related to some specific pheotypic processes, and the limited portion of the genomic data is objective to analyze. External factors that have effects on regulatory relations should be considered carefully to gain a general idea about the interactions.

Despite of these challenges, modeling such a complex dynamics gives us some crucial clues about the underlying regulatory relations. The research in this direction combined with biochemical and genetic studies will guide us to future approaches.

3.2 Continuous vs Discrete

Microarray experiments provide a finite number of experiment results, say $\bar{E}_0, \bar{E}_1, \bar{E}_2, ..., \bar{E}_{l-1}$, where $\bar{E}_m$ is a row vector representing the concentrations of all observed mRNAs at time $\bar{t}_m$, where $\bar{t}_m < \bar{t}_{m+1}$ $(m = 0, 1, 2, ..., l-2)$. Thus, the j^{th} entry in the m^{th} row vector represents the concentration of the j^{th} gene at time point $\bar{t}_m$. Given the finite set of gene expression levels, the question is how to model the dynamic nature of the gene regulations. There are mainly two modeling strategies: *time-discrete* and *time-continuos* modeling.

In [81], *Weaver et al.* explain that the current sampling times of expression data are very large so that continuous models can only be based on theoretical data. Another motivation for using discrete models is explained in [72], where *Thomas et al.* state that differential systems, a continuous approach, cannot be solved analytically but can be solved numerically to any desired precision. These motivations are the basis for most of the discrete models.

On the other hand, continuous models serve for a good approximation of the underlying regulatory network. As explained in [25], given a very large but finite number of gene expression levels, the gene expression behavior can be predicted, where gene expression levels mean training of our model from this viewpoint. In addition, repetition of the experiments and large number of experimental data with smaller sampling time strengthen the structure of the predicted behavior. Also stochastic models serve as a basis to rule out the noisy experimental data.

In the next sections, we briefly explain some of the model classes and pay attention for the continuous approaches which constitute a basis for our modeling strategy.

3.3 Current Modeling Strategies

3.3.1 Boolean Networks

Boolean networks are one of the most studied discrete approaches. The classical logical description of regulatory relations is explained by *Somogyi and Sniegoski* in

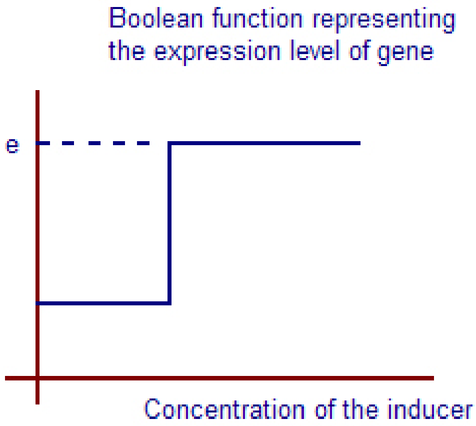


Figure 3.1: Boolean approach.

[65]. The Boolean approach represents the regulatory relations by stepwise functions. The genes in a regulatory network are assumed to be *logical variables*, i.e., their expression is either *on* or *off.* In other words, enzymes are said to be *present,* if the expression levels lie in a predefined interval which is identified by the threshold values for the enzymes and *absent* otherwise (see Figure 3.1).

The considered gene is inactive before threshold value, *e,* and is activated after the concentration achieves the threshold concentration. Then, the Boolean function related to the figure looks like:

$$w(x) := \begin{cases} 0, & \text{if} \quad x < t_i \\ 1, & \text{if} \quad x \geqslant t_i. \end{cases}$$

At a particular time, some of the genes in a cell are active while others are inactive. All active and inactive genes construct a set of Boolean variables, usually called the *metabolic state* of the cell. Gene expressions for the next time point are described by *Boolean logical rules* and updated *synchronously* in the classical description.

Liang, Fuhrman and Somogyi [48] describe an algorithm called *REVEAL* for inferring gene regulatory network from state transition table which corresponds to the time-series expression profiles. Given a finite set of gene expressions, the algorithm searches for the Boolean logical rules to infer the underlying regulation network. A small number of *state transitions*, input/output pairs is used in the algorithm, which results in a reduced search space. If the number of regulator variables for each vertex is bounded, then the algorithm identifies the network in polynomial time, but still not efficiently.

Akutsu, Miyano and Kuhara [2] propose a simpler algorithm to infer the underlying regulation network with an assumption that the number of regulatory genes are bounded by a constant. They formally define the *identification problem* as given number of variables and expression patterns, to decide whether there exists a unique Boolean network, and to give the output if it exists. The algorithm uses a simple exhaustive search for each pair of vertices and all possible Boolean functions. Unfortunately, the algorithm works less efficienctly than REVEAL and allocates more space [2]. *Akutsu et al.* [2] also state that the network cannot be identified uniquely when real expression scores are given to the algorithm, because the number of different expression patterns generated by the algorithm is so small, if the data are not random, i.e., the periodic solution problem is encountered.

Glass and Kaufmann [31] state that the Boolean network model cannot detect all the steady states in the continuous representation and the determined periodic solutions may not correspond to periodic solutions in the continuous system, when synchronous updating exist. Furthermore, they propose another approach, where the Boolean variables are updated *asynchronously. Edwards and Glass* [19] also study asynchronous updating and they use directed graphs on a bounded domain where each edge corresponds to the Boolean functions. They define the continuous Boolean function by differential equations which are piecewise linear. The existence, stability and periods of periodic orbits are deeply explained in the study.

Thomas [70, 73] introduces multiple delays associated with the different time scales and further expand the theory with *Kauffman* [71, 72]. The logical description of the model consists of two delays for each variable state change. The delays denote the time between activation signal and the response of the deactivated variable, and between deactivation signal and the response of the activated variable, respectively. The next network state is then determined according to the first possible state change. With the approach discussed, it becomes possible to detect additional logical steady states.

The theoretical approaches listed above clarify some problems, but insufficient information about the delays in the natural regulatory networks constitute a challenge to use the model for regulatory networks in nature. The Boolean approach can effectively be used for large regulatory systems and gives important clues about the biological questions. Modeling the dynamic structure and behaviour of the system with a time-discrete model has some advantages, as we discussed in Section 3.2, which are drawbacks for the Boolean network models. To overcome these challenges, some new methods have also been proposed, please refer to [54] for details where *Öktem, Pearson and Egiazarian* propose a time-continuous Boolean network with delays in the study.

3.3.2 Bayesian Networks and Statistics

Bayesian networks use statistical knowledge on analyzing the interactions between genes. Since the regulatory factors are not only the gene expression levels, other regulatory factors like protein concentrations and experimental conditions are also included in most of the Bayesian approaches. In [24], the dynamics of these factors are modelled by a directed acyclic graph, where vertices $1, 2, ..., n$ represent the regulatory factors which correspond to the random variables $X_1, X_2, ..., X_n$, respectively. These random variables are the expression levels of the genes, if the corresponding regulatory factor is a gene. *Friedman et al.* [24] define a conditional distribution $p(X_i|parents(X_i))$, where *parents(X_i)* represents the regulatory factors that have a direct influence on i, i.e., all edges that have an outgoing edge directed towards X_i. The resulting conditional distributions generate a joint probability distribution $p(X_1, X_2, ..., X_n)$. The joint probability distribution can be stated in the following way by means of *Markov assumption* (see for details [35]):

$$p(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} p(X_i|parents(X_i)).$$

Given a finite set of experimental data, Bayesian approach gives a score for the possible networks and searches for the best network which matches the given data most, or *an equivalence class* of networks (explained in [24, 34, 44]) which infer the underlying network. Although this problem cannot be solved in polynomial time, Bayesian networks give valuable results about the interactions. For example, in [24], instead of searching for a network or a class of networks, *Markov relations* and *order relations* are studied and the study determined the genes having similar functions in *S. cerevisiae,* which is studied in [66]. Please refer to [34, 41, 58] for other approaches in the literature.

Statistical analysis of the experimental data is a very effective technique to learn from the data. *Pavlidis, Lewis and Noble* [57] state that there are mainly three methodologies used in analyzing gene expressions, *unsupervised, supervised* and *semi-supervised learning* and explain the methodologies in detail. Supervised learning uses data with classification labels to train a learning algorithm and predict the unknown genes. On the other hand, unsupervised learning clusters the data based on the similarities and using classification labels generate functional clusters which are used for prediction. The last methodology, semi-supervised learning, which is described and used by *Pavlidis et al.* [57], gets the data and a set of labels to divide the data set into class groups, where these groups are then used to identify high-scoring groups.

Bayesian networks and statistical methodologies enable to deal with stochastic aspects of gene expression and noisy measurements [35, 44]. Although Bayesian networks are not computationally feasible, anticipation of the gene regulations and clustering of genes with similar functions can be easily adopted into the field.

3.3.3 Ordinary Differential Equations (ODEs)

Ordinary differential equations are deeply studied in modeling dynamical systems. The most widespread formalism to continuously model the gene regulatory systems is ODE. The differential equations consists time-dependent regulatory variables, e.g., protein and mRNA concentrations. The dynamics of the variables are modeled by rate equations for expression of any variable in terms of other variables. The equations consist of production and degradation constants for each variable in the system, introduced by *Tyson and Othmer* [76], which enables the feedback loops to be modeled.

Chen, He and Church [12] propose a dynamic system to model the gene interactions. They use two differential equations, one for the mRNA concentrations and the second for the protein concentrations. They use *Minimum Weight Solutions to Linear Equations (MWSLE)* to determine the regulatory influences. Unfortunately, the algorithm is NP-complete and does not guarantee the solution as noted in [12]. *Chen et al.* [12] also introduces another algorithm, the *Fourier Transform for Stable Systems (FTSS)* algorithm, to refine the model with cell cycle constraints.

De Hoon and Imoto [13] propose another continuous model which is similar to the described model but instead of using both mRNA and protein concentrations, only mRNA concentrations are used for approximation. *De Hoon et al.* first approximate the differential equation by a difference equation and use maximum likelihood estimation to estimate the regulatory relations, then determine the number and places of the nonzero parameters in the matrix by the so called *Akaikes Information Criterion* [35].

In a more flexible approach, *Sakamoto and Iba* [61] choose a model $\dot{E}_j = F_j(E_1, E_2, ..., E_n)$, where $j = 1, 2, ..., n$ and F_j is a function in $E_1, E_2, ..., E_n$. In other words, they assume that the change in the expression levels should be modeled by a set of functions in expression levels. *Sakamoto and Iba* find the functions F_j with the help of genetic programming combined with the least mean squares method.

Many new ideas to model the gene interaction matrices have been introduced based on these approaches, including [1, 26, 25, 28, 87]. In [26], the gene interactions are modeled by a constant gene interaction matrix, i.e., $\dot{E} = ME$ and by using least squares approximation the parameters for the regulatory relations are estimated. In this model $E \in \mathbb{R}^n$ represents a column vector, where each entry corresponds to the concentration of the corresponding gene, where there are n genes being considered. Furthermore, $\dot{E} \in \mathbb{R}^n$ denotes the column vector of change in the concentrations of genes, and $M \in \mathbb{R}^{n \times n}$ is a constant matrix whose entries correspond to the influence level of one gene to another. For example, $M_{j,i} \in \mathbb{R}^n$ denotes the influence level of the i^{th} gene to j^{th} gene.

Figure 3.2 shows the characteristics of the gene influences used in the model. If we observe this figure, we see that the influence of the inducer to the target gene is linearly constant and the concentration of the target gene increases linearly as

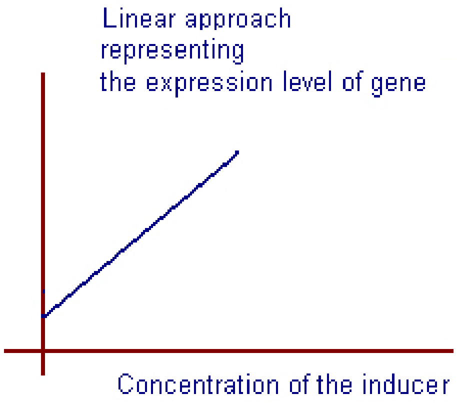


Figure 3.2: Linear approach.

the concentration of the inducer increases. In other words, if the numbers that are used to represent the inducer and the target gene are i and j, then we can say that $M_{j,i} > 0$.

The linear approach approximates the regulation curves by assuming that the rate of expression of a gene linearly increases/decreases with increasing concentration of the inducer. More precisely, in the linear approach, the interactions are expressed as linear functions. For example, the influence of $gene_i$ *to* $gene_j$ is expressed by $f_{j,i}(x) = a_{j,i}x$, where $x = E_i$ denotes the concentration of $gene_i$ and $a_{j,i} \in \mathbb{R}$. The slope of the curve gives us the interaction level, in our formula given by the coefficient $a_{j,i}$.

Since there is no predefined influences, there are n genes influencing the expression level of any gene, say $gene_j$. This implies that the change in the concentration of $gene_j$ can be approximated by the following summation which consisting of n terms:

$$F_j(E) = \sum_{i=1}^{n} a_{j,i}E_i,$$

where $E_i \in \mathbb{R}$ is the concentration of $gene_i$, as we introduced earlier.

There are also some other approaches that allow additive terms (shifts) $b_{j,i} \in \mathbb{R}$ in the linear approach. Herewith, linearity means *affine* linearity. By these additional parameters, it is possible to extend the parameter space, which is the solution space in mathematical modeling. In our model, we not only allow additive terms in linear functions, but using quadratic polynomials to represent the interactions as well.

Also some piecewise linear approaches exist in the literature [27, 29]. *Piecewise linear differential equations* are used to infer the gene regulations. For example, if the approximation consists of three linear functions with corresponding domains (a, e_1), $[e_1, e_2)$ and $[e_2, b)$, the influence of the inducer to $gene_i$ to $gene_j$ can be expressed by the following equation where for $k \in \{1, 2, 3\}$ and $(a_k)_{j,i}, (b_k)_{j,i} \in \mathbb{R}$. The piecewise linear approach is summarized in the following equation where $x = E_i$ represents the expression level of *gene i,* and there are three linear functions used to model the influence of *gene i* to *gene j*:

$$f_{j,i}(x) := \begin{cases} (a_1)_{j,i}x + (b_1)_{j,i}, & \text{if} \quad a \leqslant x < e_1 \\ (a_2)_{j,i}x + (b_2)_{j,i}, & \text{if} \quad e_1 \leqslant x < e_2 \\ (a_3)_{j,i}x + (b_3)_{j,i}, & \text{if} \quad e_2 \leqslant x \leqslant b. \end{cases}$$

In [25], a new approach is introduced based on the models described in [12, 13, 61]: $\dot{E} = M(E)E$. Here, in the right-hand side of the equation, we see that the interaction matrix M depends on the current state, E. The study does not use particular functions, e.g., linear and stepwise, for inference, instead they suggest using any model class function. They restrict the solution space to identify a unique regulatory network. For example, if the influence of $gene_i$ on the expression level of $gene_j$ is shaped like a sinus function, then the gene interaction may be formulated as, e.g., $f_{j,i}(x) := a_{j,i} \sin(b_{j,i}x)$; if there is an exponential growth in the rate of the gene expression when the inducer concentration increases, then the formula can be updated as $f_{j,i}(x) := a_{j,i}e^{b_{j,i}x}$. We remind that the characteristics related to the model function should be reflecting the dynamics in the interaction and incorporate any preinformation about the expected expression levels in mid- and long-term. In a process of *statistical learning*, where training and testing are iteratively coupled, the model class assumptions about the functions $f_{j,i}$ can step by step become improved.

Based on this approach, we model gene expression patterns by $\dot{E} = F(E)$. Here, the right-hand side $F(E)$, where $F = (F_1, F_2, ..., F_n)^T$ of our system of differential equations consists of the sum of the quadratic (constant, linear) functions:

$$F_j(E) = f_{j,1}(E_1) + f_{j,2}(E_2) + ... + f_{j,n}(E_n).$$

Here, n is the number of genes being considered. In other words, we use quadratic polynomials $f_{j,i}(x) = a_{j,i}x^2 + b_{j,i}x + c_{j,i}$, where $a_{j,i}, b_{j,i}, c_{j,i} \in \mathbb{R}$, to represent the influence of *gene i* to *gene j*. Please note that our approach represents the dynamic nature of the regulatory regulations by all constant, linear or quadratic functions, since we allow zero coefficients in the functions.

3.3.4 Other Methods

Natural language processing (NLP) and *pattern matching* have also been studied to infer the gene regulatory networks. Given textual biological data stored in databases, common motifs in the text are analyzed and the underlying gene regulatory network is inferred (for details please refer to [40, 63, 69, 86]).

The methods discussed above are all *model* formalisms. There are also *rule-based* or *knowledge-based simulation formalisms*, developed in the field of artificial intelligence [44]; please refer to [11, 30, 50] for more information.

3.4 Discussion

In [44], properties of the different models are summarized in a table. According to the table, all methods have both advantages and disadvantages. For example, Bayesian networks can be defined in time-continuous or time-discrete manners and they provide quantitative results; however, using Bayesian network is not an efficient approach for larger networks. Boolean networks are deterministic and work efficiently, but they are in the class of time-discrete models and provide qualitative results. There are many comparative studies for the models described; please refer to [18, 20, 64, 84] for details.

As described in the previous sections, the metabolic state of a cell is usually regulated by gene and enzyme regulation mechanisms in the cell, which ensure that the current metabolic state can be used to predict the next metabolic states. With this biological motivation, we state that the regulatory interactions should be modeled such that they reflect the dynamics of the system. In [1, 25], the state transition, i.e., gene interactions, matrices are functions of the current metabolic state, i.e., $M = M(E)$, where M is the gene interaction matrix and E is a vector of gene expression profiles.

We defend that $M(E)$ fits better to the experimental data than a constant matrix M. In this work, we are even going to extend that model with the system's right-hand side $M(E)E$. As we discussed above, we are going to allow nonlinear interactions and use affine linear terms, introducing shifts, there.

Aliasing is a term related to the phenomenon of a high frequency in a continuous signal masquerading as a lower frequency in the sampled output of the continuous signal. We assume that the time-intervals between neighbouring sampling times of gene expressions are sufficiently small, to prevent aliasing in the sampling of the continuous system and propose a flexible approach based on the models described in [1, 25]. The details of our model will be discussed in the coming chapters.

CHAPTER 4

MATHEMATICAL MODELING

4.1 An Overview

Given a finite number of gene expression profiles for n genes, say $\bar{E}_0, \bar{E}_1, ..., \bar{E}_{l-1}$, where each $\bar{E}_m \in \mathbb{R}^n$ is a column vector representing the gene expression profile at time $\bar{t}_m$, these times satisfying $\bar{t}_m < \bar{t}_{m+1}$ $(m \in \{0, 1, ..., l-2\})$, then the question is how to *infer* the underlying gene regulatory network at the system level. Please observe that equal time intervals are not needed for our model. In other words, in this study, our focus is on the *inverse problem* of finding the gene interactions given the experimental data for gene expressions. We model the gene interactions by means of a finite set of parameters and determine the parameter values which fit the experimental data best. Thus, our problem is a *parameter estimation problem.*

We use *ordinary differential equations (ODEs)* to model the gene regulatory networks where each regulating effect is modeled by a *quadratic polynomial* of gene expressions. Here, we note that the unknown solution is the right-hand side of our system of ODEs. This is continuous and also continuously depending on the model parameters. As stated in [1, 25], the time-continuous model describes the behavior of a continuous process in the metabolism of a cell.

In order to find the differential equations, we determine the model parameters that optimally (best) fit the experimental data based on a quadratic model ansatz. Here, we use *least squares approximation.* We underline that the success of the solution mainly depends on the size of the experimental data, i.e., whether we have an *under-determined* or *over-determined* system. The second important factor that affects the accuracy of the system consists in the errors made during experiments and approximation, *round-off* and *truncation* errors.

This chapter concerns the current approaches which use ODEs for modeling the gene regulations. Here, we also explain our model and the parameter estimation problem based on these approaches. In Chapter 5, we discuss the solution for the ODEs and also explain briefly the accuracy and stability for the system. In addition, we present a time-discrete equation and dynamics for prediction of the future behavior of our genetic system. In Chapter 6, we explain our optimization problem and introduce how we transform our system into subsystems to reduce the computational difficulty.

4.2 Ordinary Differential Equations (ODEs)

4.2.1 Linear Idealization

Chen et al. [12] model the gene interactions by the following *ordinary differential equations (ODEs)*. They use two differential equations, one for the *mRNA* concentrations and the second for the *protein* concentrations:

$$\tfrac{dr}{dt} = w(p) - Vr, \quad \tfrac{dp}{dt} = Lr - Up,$$

where t denotes the time, r and p correspond to protein and mRNA concentrations, respectively. Moreover, $w(p) = Cp$ is a linear transcription function, where C corresponds to the gene interaction matrix. In addition to this function, L denotes translational constants and V, U are constants to represent the degradation rates for *mRNAs* and *proteins*, respectively. We underline that the model is time-continuous, each variable in the equations is a function of time t. *Chen et al.* [12] propose the linear transcription model as follows:

$$\frac{dE}{dt} = ME.$$

Here, $E = (r, p)^T \in \mathbb{R}^{2n}$ represents the mRNA and protein concentrations and $M \in \mathbb{R}^{2n \times 2n}$ is the transition matrix representing regulatory interactions for both proteins and genes:

$$M = \begin{bmatrix} -V & C \\ L & -U \end{bmatrix},$$

where $C \in \mathbb{R}^{n \times n}$ represents the gene-interaction matrix where each entry $C_{j,i} \in \mathbb{R}$ ($i, j \in \{1, 2, ..., n\}$) is used to express the influence level of the expression level of $gene_i$ on the change of expression of $gene_j$. Note that *Chen et al.* use linear functions to model the gene interactions since $C_{j,i} \in \mathbb{R}$. The study mainly concerns the rate equations for the regulating factors. In the above system of differential equations, $\frac{dE}{dt} = \dot{E}$ denotes the derivative of expression functions with respect to time t:

$$\begin{bmatrix} \dot{E}_1 \\ \dot{E}_2 \\ \dot{E}_3 \\ ... \\ \dot{E}_{2n} \end{bmatrix} = \begin{bmatrix} \frac{dE_1}{dt} \\ \frac{dE_2}{dt} \\ \frac{dE_3}{dt} \\ ... \\ \frac{dE_{2n}}{dt} \end{bmatrix}.$$

By calculating the model parameters, *Chen et al.* therefore achieve a constant gene regulatory network, represented by the matrix $C \in \mathbb{R}^{n \times n}$. *De Hoon and Imoto* [13] and *Sakamoto and Iba* [61] also propose to model the gene interactions by constants. Please refer to Chapter 3 where we discussed the approaches. In other words, they all describe a linear gene regulatory network, i.e., interactions between genes are fixed and, consequently, the underlying network topology is fixed. Actually, the interactions between genes are nonlinear in nature, i.e., the influence level of one gene to another can change according to the current metabolic state.

To overcome this challenge, *Gebert, Lätsch, Pickl, Weber and Wünschiers* [25] use a more flexible approach than *Chen et al.*, *De Hoon et al.* and *Sakamoto et al.* by letting the gene regulations depend on the current metabolic state, i.e., $M = M(E)$, and they regard the following system of differential equations:

$$\dot{E} = M(E)E.$$

The metabolic state of a cell is described in terms of the gene expression levels as we discussed so far. Furthermore, in Chapter 2 we explained that the gene and enzyme regulation mechanisms prepare the cell for the next state. These continuous state transitions are an evidence for a prediction of the next state in terms of the current state. This idea sustains the fact that $M(E)$ fits better to the experimental data than a constant matrix M.

4.2.2 Polynomials for Regulatory Effects

The dynamic nature of the gene interactions should be modeled by a wide range of functions to reflect the dynamics. *Gebert et al.* [25] propose to model the gene interactions by the general case of $M(E)$ creating a need of additional restrictions on the solution space. They assume that the number of regulating factors for each gene is bounded, which corresponds to the number of incoming edges in the gene network. In other words, the number of non-zero parameters for each column of $M(E)$ is bounded.

As an inverse problem, the modeling approach does not give *a unique solution* without any restriction on the functions. We use the term "restriction on functions" because given a finite number of experimental data, the interactions can be approximated by more than one function, each having a different response for the future prediction. Unique solution to the inverse problem is an essential issue to be considered when a first principles model is not available.

In this study, we propose the following continuous equation:

$$(\mathcal{CE}) \quad \dot{E} = F(E).$$

Here, we use a tuple $F = (F_1, F_2, ..., F_n)^T$, i.e., a tuple of functions defined in the vector E of expression levels. We note that $\dot{E}(t) \in \mathbb{R}^n$ represents the change in the expression levels of genes, where n denotes the number of genes being considered. We also state that each of F_j additively comprises the functions for the change in the expression level of the $gene_j$ and it is defined in terms of $E \in \mathbb{R}^n$, i.e., the expression levels of *all* genes.

In particular, we use quadratic polynomials to model the gene interactions. Since we do not have known regulatory interactions between some particular genes, i.e., activating genes for some gene and the genes that are activated by that gene are not known, we have in total n^2 polynomials which generate a set of gene interaction polynomials. For example, the influence of $gene_i$ on $gene_j$ is represented by a quadratic polynomial, say $f_{j,i}(x) = a_{j,i}x^2 + b_{j,i}x + c_{j,i}$, where $x = E_i$ denotes the concentration of $gene_i$ and $a_{j,i}, b_{j,i}, c_{j,i} \in \mathbb{R}$.

Consequently, the increase/decrease in the expression level of $gene_j$ is represented by $F_j(E)$, determined by the following summation:

$$F_j(E) = \sum_{i=1}^{n} f_{j,i}(E_i) = \sum_{i=1}^{n} (a_{j,i}(E_i)^2 + b_{j,i}(E_i) + c_{j,i}).$$

Our model class extension by additional constant terms on the right-hand side of our system can also be represented by the introduction of an additional *shift* vector C on the right-hand side of the continuous equation, $\dot{E} = M(E)E$, which is proposed by [25]. Without any constant terms, $0 \in \mathbb{R}^n$ always belongs to the stationary points. In our model, 0 can but needs not to be a stationary point. This shows that our extended modeling also implies an extension in the range of structures of trajectory sets. Please note that these constant terms represent the environmental effects and other constraints which affect the gene expression levels in the environment.

Our approach is pioneering and shall serve for a next step in the understanding, after approaches by linear functions $f_{j,i}(x) = a_{j,i}x$ [12, 13, 26, 61]. Quadratic polynomials extend the solution space when compared to the linear idealization and produce more accurate results than the linear models. As a model class, quadratic polynomials include the linear polynomials, which is an advantage of our quadratic approach.

For future research, we recommend and plan further refined investigation with extended classes of polynomials, splines, trigonometric, exponential functions and suitable combinations of these functions such that the ODEs are solvable. Up to multiplying in the right-hand side with E, which incorporates some kind of a trend factor, for example, trigonometric functions represent cyclical (e.g., seasonal) components, while splines are a time-interval wise refinement of representing the process polynomially.

Jacob and Monod [52] has observed that gene activation or inhibition takes place when the corresponding protein concentration exceeds a threshold. This behavior can be formulated by using switching local polynomials where the state transition function is a polynomial, but the parameters of this polynomial may switch when a relevant threshold is exceeded. In fact, such a formulation will add the capability of handling multistationary dynamics to our model. Then, it will be possible to involve differentiation, cell specialization and adaption into the model. Finding out the methods to infer the system in such a model class is in our future research plan.

CHAPTER 5

FUTURE PREDICTION

5.1 Discrete Equation and Dynamics

A numerical solution for an ODE provides a finite set of values of the solution function which reflects the behavior of the cell metabolism identified by the differential equation. Let us refer to the system $(\mathcal{CE})$. For a particular state, the next states are generated iteratively, i.e., $\hat{E}(t_k)$, $\hat{E}(t_{k+1}), ..., \hat{E}(t_{s-1})$ are approximated in consecutive order, where $k, s \in \mathbb{N}$. Given $E(t_0)$, some initial state, we start with the initial state $\hat{E}(t_0) = E(t_0)$ and predict the next state which is then used to predict the next states. In our examples which will be given in the next chapters, we use $\hat{E}(t_0) = \bar{E}_0$, the first expression profile given by the experiments as the initial state.

In [25], depending on the model described in the following sections, a time-discrete equation and dynamics is represented as follows:

$$\hat{E}_{k+1} = \mathbb{M}_k \hat{E}_k \quad (k \in \mathbb{N}_0),$$

where $\mathbb{M}_k := \ I + h_k M(\hat{E}_k)$, I is the $n \times n$ unit matrix and $h_k := t_{k+1} - t_k$.

Here, h_k represents the time difference between two consecutive approximations. In our examples, we take $h_k = \bar{h}_m$ $(k = m \in \{0, 1, 2, ..., l-2\})$ for the approximation part to be able to compare the experimental data with approximated expression levels, i.e., to be able to compare $\bar{E}_0, \bar{E}_1, \bar{E}_2, ..., \bar{E}_{l-1}$ with $\hat{E}_0, \hat{E}_1, \hat{E}_2, ..., \hat{E}_{l-1}$. Please note that the time differences between approximations can be adjusted according to the underlying biological motivation. In our work, we take $\bar{h}_m = 1$ $(m \in \{0, 1, 2, ..., l-2\})$ and $h_k = 1$ $(k \in \{0, 1, 2, ..., l-1\})$. In other words, we approximate the experimental data with our time-discrete equation and also predict the next state, i.e., $\hat{E}_l$.

Very analogously from the viewpoint of definition of the time-discrete dynamics, we present the following discrete equation:

$$(\mathcal{DE}) \quad \hat{E}_{k+1} = \mathbb{F}_k(\hat{E}_k) \quad (k \in \mathbb{N}_0),$$

where $\mathbb{F}_k = ((\mathbb{F}_k)_1, (\mathbb{F}_k)_2, ..., (\mathbb{F}_k)_n)^T$ is a tuple of functions defined in terms of the expression levels of the considered genes for the k^{th} time step, i.e., $\hat{E}_k$, such that:

$$(\mathbb{F}_k)_j(\hat{E}_k) := \; (\hat{E}_k)_j + h_k \sum_{i=1}^{n} f_{j,i}((\hat{E}_k)_i).$$

Please note that in the given formula $(\hat{E}_k)_j$ denotes the concentration of $gene_j$ at the current state, i.e., at k^{th} time step. In this definition, functions $f_{j,i}(x)$ are the influence functions defined in Chapter 4. We remind that in the linear case, the coefficients of the linear functions corresponds to the entries of the matrix M. However, in our model, for each time step we have a distinct set of functions $F_1, F_2, ..., F_n$.

In this general approach, which basically comes from statistical learning, we infer the model from empirical observation which is already sampled, so we do *not* discretize the system. Based on [25], in the next section, we introduce Euler's Method mainly for conceptual reasons. In fact, higher frequency components of the gene expression profile which might cause instability of the numerical solutions, are already lost during sampling due to an effect called *aliasing* (please refer to Section 3.4). However, it is a completely instrumentation related problem and, therefore we had to ignore such effects.

5.2 Euler's Method

Euler's Method is derived from *Taylor's Theorem* which states:

Suppose $p \in \mathbb{N}$, that E and its derivatives $E^{'}, E'', ..., E^{(p-1)}$ are defined and continuous on $D = [a, b]$, and that $E^{(p)}$ exists in (a, b). If $t, t + h \in D$, then there exists a number $\gamma \in (t, t+h)$ (if $h > 0$) or $\gamma \in (t+h, t)$ (if $h < 0$) such that:

$$E(t+h) = E(t) + E^{'}(t)h + \frac{1}{2!}E^{''}(t)h^2 + ... + \frac{1}{(p-1)!}E^{(p-1)}(t)h^{p-1} + \frac{1}{p!}E^{(p)}(\gamma)h^p.$$

The proof for the theorem is given in [8].

Euler's method uses Taylor's Theorem for $p = 2$. In case of existence of infinitely often differentiability and local convergence, our function can locally be represented by a power series. More precisely, Euler's Method [43] approximates the solution value by the following equation, where $\hat{E}$ and $\hat{E}'$ denote the approximated expression scores and approximated change in the expression scores and $h_k := t_{k+1} - t_k$:

$$\hat{E}(t_{k+1}) = \hat{E}(t_k) + \hat{E}'(t_k)h_k.$$

Let us denote $\hat{E}_k := \hat{E}(t_k)$. Furthermore, use $\hat{E}'(t_k) = F(\hat{E}_k)$, which we obtained for the approximation to the derivative. Thus, we get:

$$\hat{E}_{k+1} = \hat{E}_k + F(\hat{E}_k)h_k.$$

We define $\mathbb{F}_k = ((\mathbb{F}_k)_1, (\mathbb{F}_k)_2, ..., (\mathbb{F}_k)_n)^T$ with $(\mathbb{F}_k)_j$ just as we have written in the previous section. Then, we obtain the following time-discrete equation and dynamics for all $k \in \mathbb{N}_0$:

$$\hat{E}_{k+1} = \mathbb{F}_k(\hat{E}_k).$$

The discrete equation and dynamics enables us to find the next states, given an initial state $\hat{E}(t_0)$. We note that the recursive definition of the next states show that, for the $(k+1)^{th}$ approximation, we get $\hat{E}_{k+1} = \mathbb{F}_k(\mathbb{F}_{k-1}(\mathbb{F}_{k-2}(...\mathbb{F}_0(\hat{E}_0)...)))$. Here, let us also state the recursive definition for the model $\dot{E} = M(E)E$, to allow an insight into the difference between this model and our model. We see that given the initial state, i.e., $\hat{E}_0$, the $(k+1)^{th}$ state is defined in the following way: $\hat{E}_{k+1} = \mathbb{M}_k\mathbb{M}_{k-1}\mathbb{M}_{k-2}...\mathbb{M}_0\hat{E}_0$.

This special structure enables us to study boundedness of the solution easily. The following sections show the accuracy and stability issues for our problem. We always refer to the linear and nonlinear functions together to show the similarities and differences between approaches. The following sections also explain both cases.

5.3 Accuracy of the System

Accuracy measures the approximation error between actual and computed solution. There are two main sources which increase the approximation error: *rounding* and *truncation* errors.

Finite precision of floating-point arithmetic generates the rounding error. Computers store a real number in a finite number of bytes and, if necessary, round the number. Rounding of a real-number results in a rounding error. For example, numbers smaller than machine precision are rounded to zero. We reduce the rounding-error by some methods discussed in [36].

On the other hand, the source for truncation error is the method used in discretization. There are two kinds of truncation errors: *local and global truncation errors*. The local truncation error is defined as $L_k = \hat{E}_k - u_{k-1}(t_k)$ $(k \in \mathbb{N}\backslash\{1\})$, where $\hat{E}_k$ is the computed solution at time t_k and u_{k-1} is the solution curve determined for the previous time step, t_{k-1}. The global truncation error is the difference between the computed solution and the true solution determined the initial solution curve u_0 and stated as: $G_k = \hat{E}_k - u_0(t_k)$ $(k \in \mathbb{N})$. Global truncation errors are defined in [36], and they show the error, made in one step of the numerical method, and the difference between the computed and true solution determined by the initial state. A numerical method is said to be *first-order accurate* if $L_k = O(h_k^2)$. Euler's method is first-order accurate; please refer to [36] for the proof.

There is a *trade-off* between the rounding and truncation error when we consider the sample times. In order to continuously approximate the given experimental data effectively, we need a very small *sampling time*, $\bar{h}_m$ $(m \in \{0, 1, 2, ..., l-2\})$, but a small sampling time generates rounding errors. On the other hand, a large sample time produces worse results than the latter case but results in a lowered rounding error. A suitable experimental sample time should be decided according to the current considerations about the model and the system availability.

5.4 Stability and Boundedness of the System

Stability theory, which studies the sensitivity of a solution of an ODE with respect to perturbations, is described in [36]. If the responses of solution curves for an ODE gets closer with time, a small perturbation to a solution will shrink (or essentially remain of the same order) within time, i.e., the system is *stable.* On the other hand, *instability* of an equation means that a small perturbation to a solution will grow within time since the members of the solution family for an ODE move away from each other with time. Another stability term is *balanced (neutrally stable)*, which means that there is no convergence or divergence of solution curves within time.

The stability concept of an ODE depends on the entire family of solutions, not on a particular solution [36]. For $\dot{E} = F(E) = ME$, the eigenvalues of the following *Jacobian matrix* $J = (J_{i,j})$ shows the stability/instability of the system:

$$J_{i,j} = \frac{\partial F_i}{\partial E_j}.$$

This type of stability is the most widely accepted stability criterion. Sometimes, it is referred to as *Lyapunov stability,* to distinguish if any other stability criterion is also used [77, 78].

In [28], stability in the theory of ODEs is referring to the case of autonomous systems, where the right hand-side of $F(E) = M(E)E$ does not depend on t [56]. Furthermore, in [4, 28, 56], the analytic definition of stability refers to stationary points E^*, where $F(E^*) = 0$.

5.4.1 Stability for a Linear System

Observe that, for $M \in \mathbb{R}^{n\times n}$, the Jacobian matrix, J is equal to M. For such a system, the negativity of the eigenvalues of the matrix M implies the stability of the system. Let the eigenvalues of M be $\delta_1, \delta_2, \delta_3, ..., \delta_n$, and let us focus on the case, where all these values are real numbers, i.e., $\delta_i \in \mathbb{R}$ $(i \in \{1, 2, ..., n\})$. This is, e.g., guaranteed in the case where M (or, lateron, J) is symmetric: $M = M^T$ (or, in general nonlinear case, if, e.g., we have a so-called gradient flour). If, however, there is an eigenvalue with nonvanishing imaginary point, i.e., one (or more) $\mathrm{Im}(\delta_{i'}) \in \mathbb{C} \backslash \mathbb{R}$, then we would study the signs of the eigenvalues' real parts $\mathrm{Re}(\delta_i)$ $(i \in \{1, 2, ..., n\})$ in the sequel instead. If any eigenvalue is positive, then the equation is unstable. If all the eigenvalues are negative, then the equation is stable. Finally, a neutrally stable equation exists when one or more eigenvalues are zero and all other eigenvalues are negative. The following table illustrates stability and unstability conditions:

$$\begin{array}{llll} \text{if} & \forall i \in \{1,2,...,n\} & \delta_i < 0 & \text{->} \quad \text{stable system,} \\ \text{if} & \exists i \in \{1,2,...,n\} & \delta_i > 0 & \text{->} \quad \text{unstable system.} \end{array}$$

Let us consider a gene interaction matrix for two genes:

$$M = \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} \in \mathbb{R}^{2\times 2}.$$

For any $E = \begin{bmatrix} E_1 \\ E_2 \end{bmatrix}$ we have $F = \begin{bmatrix} a_{1,1}E_1 + a_{1,2}E_2 \\ a_{2,1}E_1 + a_{2,2}E_2 \end{bmatrix}$, which is used to determine:

$$J = \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix}.$$

All entries of the Jacobian matrix are constant values and correspond to the entries in the gene interaction matrix. Please note that, for a linear idealization, estimated parameters show the stability/instability of the system. In addition, in linear time-invariant case, the system is stable for all states if it is stable for any state and this also corresponds to boundedness of solutions. In other words, *Bounded Input and Bounded Output (BIBO)* stability [55].

5.4.2 Stability for a Nonlinear System

When the system is nonlinear, it might be unstable for some range of the state space while it is stable for the rest [3]. In fact, such nonlinear system structures are necessary for regulatory dynamics [1, 54, 71, 72].

Stability analysis for $M = M(E)$ needs more effort than in the constant case. The stability analysis of such a system gives a parametrical boundary for the stable solution, since we search for the values of parameters that satisfy the stability criterion, i.e., a subspace that satisfies the negativity of the eigenvalues of the Jacobian matrix.

Let $f_{j,i}(x) = a_{j,i}x^2 + b_{j,i}x + c_{j,i}$ be a function that corresponds to the i^{th} additive contribution in F_j matrix, where $x = E_i$ denotes the concentration of $gene_i$ and $a_{j,i}, b_{j,i}, c_{j,i} \in \mathbb{R}$. Then, we have the following F vector and J matrix:

For any $E = \begin{bmatrix} E_1 \\ E_2 \end{bmatrix}$, we have

$$F = \begin{bmatrix} (a^2_{1,1}E_1 + b_{1,1}E_1 + c_{1,1}) + (a_{1,2}E_2^2 + b_{1,2}E_2 + c_{1,2}) \\ (a_{2,1}E_1^2 + b_{2,1}E_1 + c_{2,1}) + (a_{2,2}E_2^2 + b_{2,2}E_2 + c_{2,2}) \end{bmatrix},$$

which we use to determine the Jacobian matrix $J = J(E)$:

$$J = \begin{bmatrix} 2a_{1,1}E_1 + b_{1,1} & 2a_{1,2}E_2 + b_{1,2} \\ 2a_{2,1}E_1 + b_{2,1} & 2a_{2,2}E_2 + b_{2,2} \end{bmatrix}.$$

Here, the estimated parameters are not the only considerations for stability criteria. The eigenvalues of the Jacobian matrix, which are functions of E_1 and E_2, are used to analyze the stable and unstable regions for the solution.

Stability analysis for *numerical* methods is different from the described analysis. The time-discrete system for the model $\dot{E} = M(E)E$ is stable if the absolute value of the eigenvalues of all considered matrices $\mathbb{M}_i$ ($i \in \{0, 1, 2, ..., q-1\}$) and of any finite product of them are smaller than 1 [10, 28, 56]. Here, q is the number of considered approximative matrices. Please note that a similar spectral analysis can be made for our nonlinear (quadratic) case.

5.4.3 BIBO Stability and Boundedness

When we use linear idealization, the stability of the system implies the boundedness of the solution since the matrices in each iteration are constant and equal to the gene interaction matrix M. An extremely useful criterion for nonlinear systems is BIBO stability, meaning that any bounded input (in our case, a perturbation) to the system will result in a bounded deviation of the solutions [55].

For $(\mathcal{CE})$, given an initial state, the discrete equation $(\mathcal{DE})$ presented above, calculates the next state recursively, i.e., the right-hand side of the equation comes from a matrix multiplication of gene interaction matrices calculated at previous states and the initial state. Let us at each step k consider a set of matrices $\mathbb{M} = \{\mathbb{M}_0, \mathbb{M}_1, \mathbb{M}_2, ..., \mathbb{M}_{q-1}\}$, which we regard to finitely approximate the corresponding continuous range of matrices. The cardinality of this set, q, needs not but can be adopted, when k increases. In our notation of the set $\mathbb{M}$, the index ν of the matrices $\mathbb{M}_\nu$ must not be the iteration index k for the first steps.

For the k^{th} step, if the absolute values of the eigenvalues of any finite product of matrices from $\mathbb{M}$ are smaller than 1, then this implies that we have a bounded solution. In [25, 28], this kind of stability provided by these matrix products, i.e., by the linear mappings defined by the elements of $\mathbb{M}$ is analyzed by an algorithm in [10, 59] which is detecting stability. However, a recent trend is explicitly stating if any criterion else than *Lyapunov stability* is used.

CHAPTER 6

OPTIMIZATION PROBLEM

We described our model in Chapter 4 where we explained the system of ODEs and discussed the characteristics of quadratic polynomials to model the regulatory relations. Then, in Chapter 5, our focus was on the solution for the ODE where we also gave a brief discussion about the stability and accuracy of the system. This chapter concerns the optimization problem to solve the model parameters, where we also discuss the uniqueness of the solution. Because of the underlying biological motivation, i.e., gene networks are very large networks, we show a way to decompose our optimization problem into subproblems which significantly reduces the computation time.

6.1 Fundamental Approaches

In [1, 25], *Akhmet et al.* and *Gebert et al.* introduce nonlinear regulatory relations and propose the model: $\dot{E} = M(E)E$. The system is solved by optimization techniques. While explaining their approach, firstly, they determine the regulatory relations with a constant regulatory matrix. We find the idea compatible to facilitate in describing our approach. They determine the model parameters, i.e., the entries in regulatory matrix $M \in \mathbb{R}^{n \times n}$, by the least squares method. The minimization problem is then transformed to the well-known canonical form to achieve the availability of the least squares approaches for this form as shown in [1, 25].

An important note from statistical learning should be stated here: The parameters used in the optimization problem are divided into two parts in these studies. The first set of parameters consists of the expression metabolism parameters which are in the second step used to analyze the boundedness (stability) of the system. The remaining parameters are in the first step used for modeling. Please note that the parameters of the second set play a similar role as *training data,* while the parameters in the first set are similar to *test data* (please refer to [35] for details). A stable solution for the optimization problem is then used to infer the underlying regulatory network.

6.2 Existence and Uniqueness

Inverse problems are known to be hard problems since given the experimental data, it is usually hard to obtain the exact model that satisfies the model constraints among all models which adequately fit the given data [5]. On the other hand, there may be no model which exactly fits the data if the given data contain high experimental noise. Our least squares problem has at least one solution $\hat{M}$ because of the continuous dependence of the nonnegative objective function on M and because of this objective function's (generically) quadratic growth. Now, the uniqueness of the solution should be considered for an inverse problem. In the next paragraphs, we discuss this *uniqueness of the solution.*

If polynomials are used for regulatory relations, the number of unknowns for each regulatory relation is equal to the degree of the polynomial, say p. Since there are n^2 regulatory polynomials, for such a system, there are in total $(p+1)n^2$ unknowns for the optimization problem. In our quadratic approach, for each regulatory quadratic polynomial $f_{j,i}$, we have three unknowns corresponding to the coefficients of the polynomial, namely $a_{j,i}, b_{j,i}, c_{j,i}$. Thus we have in total $3n^2$ unknowns. On the other hand, the number of knowns is equal to the size of the experimental data, i.e., l. For such a system, the uniqueness of the solution depends on all: p, n and l, i.e., on their constellation, and on the rank of the *system matrix* which we get when turning our approximation problem into the form of n subproblems of canonical linear least squares form. We shall come to the subproblems very soon.

If the number of knowns is smaller than the number of unknowns, then the system is an *under-determined system.* Under-determined systems do not have a unique solution. Unavailability of the current experimental data forces us to restrict the solution space to have a unique solution for our under-determined system. Here, we should note that repetition of the experiments and a wide range of experimental data combined with smaller sampling times will help us in determining the model parameters and obtaining a high accurate system.

However, when we have a linear system, if the number of knowns and unknowns are equal and if we have a full-rank matrix, there is a unique solution. Furthermore, if the number of knowns is even larger, i.e., *over-determined system*, then we search for the curve that fits to the experimental data best. Given an accuracy criterion, we can numerically search for the curve that fits the data until the needed accuracy value is reached. Because of the growth reasons (for the forthcoming n objective functions), there is at least one solution for such a problem. Please refer to *Isaacson and Keller* [42] for analytical discussions on numerical solutions.

In this study, we use the optimization problem mentioned above where all model parameters are assumed to be used in gene regulations. Then, we discuss the family of regulatory polynomials and transform the system to a new form where the optimization problem is equivalently divided into n subproblems. For each subproblem, we have in total $3n$ unknowns and l knowns.

6.3 Approximation to the Derivative

As we noted earlier, experiments provide us a finite set $\{\bar{E}_0, \bar{E}_1, ..., \bar{E}_{l-1}\}$, where $\bar{E}_m \in \mathbb{R}^n$ is the experiment result given at time $\bar{t}_m$, where $\bar{t}_m < \bar{t}_{m+1}$ and $m \in \{0, 1, 2, ..., l-2\}$. In addition to this information, using *a finite difference quotient*, we get an approximation of the left hand-side of the algebraic equation $\dot{E}(t) = M(E)E$ as follows:

$$\dot{\bar{E}}_m := \begin{cases} \frac{\bar{E}_{m+1} - \bar{E}_m}{\bar{t}_{m+1} - \bar{t}_m}, & \text{if} \quad 0 \leqslant m < l-1 \\ \frac{\bar{E}_m - \bar{E}_{m-1}}{\bar{t}_m - \bar{t}_{m-1}}, & \text{if} \quad m = l-1. \end{cases}$$

6.4 Revised Optimization Problem

6.4.1 Linear Approach

For a static regulatory network, the model parameters, i.e., the entries in the regulatory matrix $M \in \mathbb{R}^{n \times n}$, are determined by the least squares method which has the form:

$$\min_{M=(M_{j,i})} \sum_{m=0}^{\ell-1} ||M\bar{E}_m - \dot{\bar{E}}_m||^2 \,.$$

Please note that this problem has at least one solution $\hat{M}$. Here, $\bar{E}_m$ and $\dot{\bar{E}}_m$ are column vectors for the experimental data and difference quotients of the expression levels of n genes at time $\bar{t}_m$ for $m \in \{0, 1, 2, ..., l-1\}$, and $||.||$ is the Euclidian norm.

The idea comes from the fact that, the increase/decrease in the expression level of a gene, say $gene_i$ can be approximated by the sum of products of influence factors for $gene_i$ and expression levels of influencing genes. More precisely, we have the following equality for all $j \in \{1, 2, ..., n\}$ and for all $m \in \{0, 1, ..., l-1\}$:

$$(\dot{\bar{E}}_m)_j = (M_{j,1}(\bar{E}_m)_1 + M_{j,2}(\bar{E}_m)_2 + M_{j,3}(\bar{E}_m)_3 + ... + M_{j,n}(\bar{E}_m)_n) + (AE_m)_j.$$

Here, $(AE_m)_j$ is used to express the *approximation error* for the change in the concentration of $gene_j$ in the m^{th} experiment time point and $M_{j,i} \in \mathbb{R}$ denotes the influence of the expression level of $gene_i$ to the rate of trancription of $gene_j$, being an entry of the following matrix:

$$M = \begin{bmatrix} M_{1,1} & M_{1,2} & M_{1,3} & ... & M_{1,n} \\ M_{2,1} & M_{2,2} & M_{2,3} & ... & M_{2,n} \\ M_{3,1} & M_{3,2} & M_{3,3} & ... & M_{3,n} \\ .. & ... & ... & ... & ... \\ M_{n,1} & M_{n,2} & M_{n,3} & ... & M_{n,n} \end{bmatrix}.$$

Please note that the parameters estimated by the main problem are equal to the parameters which are estimated by means of these subproblems. Since there is no coupling between the equations corresponding to any two genes, say $gene_{j_1}$ and $gene_{j_2}$ ($j_1 \neq j_2$), the minimization problem can in fact *equivalently* be separated into n subproblems, each including l approximations as follows, for each $gene_j$:

$$\begin{aligned}
(\dot{\bar{E}}_0)_j &= (M_{j,1}(\bar{E}_0)_1 + M_{j,2}(\bar{E}_0)_2 + ... + M_{j,n}(\bar{E}_0)_n) + (AE_0)_j, \\
(\dot{\bar{E}}_1)_j &= (M_{j,1}(\bar{E}_1)_1 + M_{j,2}(\bar{E}_1)_2 + ... + M_{j,n}(\bar{E}_1)_n) + (AE_1)_j, \\
&... \\
(\dot{\bar{E}}_{l-2})_j &= (M_{j,1}(\bar{E}_{l-2})_1 + M_{j,2}(\bar{E}_{l-2})_2 + ... + M_{j,n}(\bar{E}_{l-2})_n) + (AE_{l-2})_j, \\
(\dot{\bar{E}}_{l-1})_j &= (M_{j,1}(\bar{E}_{l-1})_1 + M_{j,2}(\bar{E}_{l-1})_2 + ... + M_{j,n}(\bar{E}_{l-1})_n) + (AE_{l-1})_j.
\end{aligned}$$

Now, our subproblem associated to $gene_j$ looks as follows: We want to minimize the approximation errors with respect to the entries of $(M^T)_j$, i.e., j^{th} row vector of M written as a column vector:

$$\min_{(M^T)_j} \sum_{m=0}^{\ell-1} (\bar{E}_m^T (M^T)_j - (\dot{\bar{E}}_m)_j)^2 \,,$$

where M^T denotes the transpose of the matrix M. Furthermore, $\bar{E}_m^T$ is used to express the transpose of the vector $\bar{E}_m$. Please note that the approximated change in the expression levels and expression levels at time point $\bar{t}_m$ are represented by $\bar{E}_m$ and $\dot{\bar{E}}_m$ in this formula. All of these n subproblems are of *canonical* linear least squares form, i.e., they have a vectorial unknown.

Let us summarize the linear approach: Given a finite set of expression levels, $\{\bar{E}_0, \bar{E}_1, \bar{E}_2, ..., \bar{E}_{l-1}\}$, the finite difference quotients give us another finite set of the approximate increase/decrease in the expression levels, $\left\{\dot{\bar{E}}_0, \dot{\bar{E}}_1, \dot{\bar{E}}_2, ..., \dot{\bar{E}}_{l-1}\right\}$. We search for the most accurate matrix M that minimizes the sum of squares of differences between approximated and given data. As we noted in the previous chapter, estimates for the next states are given by the iteration procedure $\hat{E}_{k+1} = \mathbb{M}_k \hat{E}_k$ $(k \in \mathbb{N}_0)$, where $\hat{E}_k$ denotes the approximated expression scores for the genes at the corresponding time. This is a time-discrete dynamics, based on the initial value $\hat{E}_0$, e.g., $\hat{E}_0 = \bar{E}_0$. The following example illustrates the optimization problem.

Example 6.4.1.1

Let us analyze the response of our model for randomly generated sample, where the number of genes being observed is $n = 3$ and the number of time points as $l = 5$ given in Table 6.1. As you observe from this table, the entries in the first row correspond to the entries of the vector $\bar{E}_0$. For example, the concentration of the second gene at the initial time point is expressed by $(\bar{E}_0)_2 = 0,9726$.

		$gene_1$	$gene_2$	$gene_3$			
at $\bar{t}_0$	[	0,7778	0,9726	0,7802	]	=	$\bar{E}_0$
at $\bar{t}_1$	[	0,2324	0,6999	0,1815	]	=	$\bar{E}_1$
at $\bar{t}_2$	[	0,7026	0,9657	0,0424	]	=	$\bar{E}_2$
at $\bar{t}_3$	[	0,0032	0,8392	0,0503	]	=	$\bar{E}_3$
at $\bar{t}_4$	[	0,4908	0,0975	0,2250	]	=	$\bar{E}_4$

Table 6.1: *Expression scores of three genes on five time points.*

We have in total three genes and nine interaction functions between these genes. So, for this particular problem, the gene interaction matrix $M \in \mathbb{R}^{3\times3}$ generally looks as follows:

$$M = \begin{bmatrix} M_{1,1} & M_{1,2} & M_{1,3} \\ M_{2,1} & M_{2,2} & M_{2,3} \\ M_{3,1} & M_{3,2} & M_{3,3} \end{bmatrix}.$$

Assuming the difference between two consecutive time points, $\bar{h}_m = 1$ for $m = 0, 1, ..., l-2$, we use finite difference quotients and get the approximate increase and decreases in the expression levels shown in Table 6.2.

		$gene_1$	$gene_2$	$gene_3$			
at $\bar{t}_0$	[	$-0,5454$	$-0,2727$	$-0,5987$	]	=	$\dot{\bar{E}}_0$
at $\bar{t}_1$	[	$+0,4702$	$+0,2658$	$-0,1391$	]	=	$\dot{\bar{E}}_1$
at $\bar{t}_2$	[	$-0,6994$	$-0,1265$	$+0,0079$	]	=	$\dot{\bar{E}}_2$
at $\bar{t}_3$	[	$-0,4876$	$-0,7417$	$+0,1747$	]	=	$\dot{\bar{E}}_3$
at $\bar{t}_4$	[	$-0,4876$	$-0,7417$	$+0,1747$	]	=	$\dot{\bar{E}}_4$

Table 6.2: *Approximation to the changes in the expression levels.*

We explained that since there is no coupling between the expression functions of any two genes, we can equivalently divide the problem into n subproblems. Please observe that the following least squares problem represents the j^{th} subproblem, i.e., the optimization problem for $gene_j$ by using the Euclidian norm $\|.\|$:

$$\min_{M_{j,1},M_{j,2},M_{j,3}} \left\| \begin{bmatrix} 0,7778 & 0,9726 & 0,7802 \\ 0,2324 & 0,6999 & 0,1815 \\ 0,7026 & 0,9657 & 0,0424 \\ 0,0032 & 0,8392 & 0,0503 \\ 0,4908 & 0,4908 & 0,2250 \end{bmatrix} \begin{bmatrix} M_{j,1} \\ M_{j,2} \\ M_{j,3} \end{bmatrix} - \begin{bmatrix} (\dot{\bar{E}}_0)_j \\ (\dot{\bar{E}}_1)_j \\ (\dot{\bar{E}}_2)_j \\ (\dot{\bar{E}}_3)_j \\ (\dot{\bar{E}}_4)_j \end{bmatrix} \right\|^2 .$$

Let a solution of this problem be expressed as $(\hat{M}^T)_j = \begin{bmatrix} \hat{M}_{j,1} & \hat{M}_{j,2} & \hat{M}_{j,3} \end{bmatrix}^T$. The least squares problem is solved by *lsqlin* function provided by *MATLAB*. *lsqlin* solves a linear system in the least squares sense and outputs the solution vector, the residual and the residual norm. For the subproblem we presented, the solution vector is $(\hat{M}^T)_j$ and the *residual vector* $\hat{R}_j = ((\hat{R}_j)_0, (\hat{R}_j)_1, ..., (\hat{R}_j)_{l-1})^T$ is defined as follows:

$$\hat{R}_j = \begin{bmatrix} \bar{E}_0^T(\hat{M}^T)_j - (\dot{\bar{E}}_0)_j \\ \bar{E}_1^T(\hat{M}^T)_j - (\dot{\bar{E}}_1)_j \\ \bar{E}_2^T(\hat{M}^T)_j - (\dot{\bar{E}}_2)_j \\ ... \\ \bar{E}_{l-1}^T(\hat{M}^T)_j - (\dot{\bar{E}}_{l-1})_j \end{bmatrix},$$

and shows the least squares approximation error. Furthermore, the *residual norm* is the norm of the residual vector:

$$norm(\hat{R}_j) = \sum_{m=0}^{l-1}((\hat{R}_j)_m)^2.$$

For each gene, we solved the least squares problem for this particular example and obtained the following gene regulatory matrix for this particular problem:

$$\hat{M} = \begin{bmatrix} -0,8138 & +0,2048 & +0,2926 \\ -0,2018 & -0,1542 & -0,0889 \\ +0,1744 & +0,0188 & -0,8617 \end{bmatrix}.$$

After determining the gene regulatory matrix, we can conclude about the gene interactions. In the previous chapters, we expressed the linear regulatory function in the following way $f_{j,i}(x) = a_{j,i}x$, where $x = E_i$ denotes the expression level of *gene*$_i$. After we solve the model parameters, i.e., the coefficients of the functions, we observe that $a_{j,i} = \hat{M}_{j,i}$. For example, we denoted the influence of *gene$_1$* to *gene$_2$* by $f_{2,1}(x) = a_{2,1}x$, where x denotes the expression level of *gene*$_1$. After we solved the least squares subproblem for the *second* gene, we determined that $a_{2,1} = -0,2018$, $a_{2,2} = -0,1542$ and $a_{2,3} = -0,0889$.

In Chapter 5, we explained that the next expression level of the *gene*$_j$ can be approximated by the following equation:

$$(\hat{E}_k)_j + h_k \sum_{i=1}^{n} f_{j,i}((\hat{E}_k)_i),$$

where k denotes the current step, h_k is the difference between two consecutive time points and $\hat{E}_k$ is an approximation vector of expression levels at time point k. Suppose the initial expression levels are equal to the first expression level vector provided by the experiment, i.e., $\hat{E}_0 := \bar{E}_0$. Also, suppose that the time difference between any two approximations is 1, i.e., $h_k = t_{k+1} - t_k = 1$ $(k \in \{0, 1, ..., s-2\})$. Here, $s \in \mathbb{N}_0$ denotes the chosen number of iterations to approximate and predict the next states given the initial state. In fact, as we stated earlier, to be able to compare the experimental data with the approximated data, we take $h_k = 1$, since the time difference between the experimental data points are 1. However, please observe that, in general, the time differences between two consecutive approximations need *not* to be the same as the time difference between neighboring sampling times.

For example, the next expression level for *gene$_2$* can in our example be calculated

$$\begin{array}{lcccccccc}
 & & gene_1 & gene_2 & gene_3 & & & \\
\text{at } t_0 & [& 0{,}7778 & 0{,}9726 & 0{,}7802 &] & = & \hat{E}_0 \\
\text{at } t_1 & [& 0{,}5723 & 0{,}5963 & 0{,}2619 &] & = & \hat{E}_1 \\
\text{at } t_2 & [& 0{,}3053 & 0{,}3655 & 0{,}1473 &] & = & \hat{E}_2 \\
\text{at } t_3 & [& 0{,}1748 & 0{,}2344 & 0{,}0805 &] & = & \hat{E}_3 \\
\text{at } t_4 & [& 0{,}1041 & 0{,}1558 & 0{,}0460 &] & = & \hat{E}_4 \\
\text{at } t_5 & [& 0{,}0648 & 0{,}1067 & 0{,}0275 &] & = & \hat{E}_5
\end{array}$$

Table 6.3: *Approximation with linear functions.*

as follows:

$$\begin{aligned}
(\hat{E}_1)_2 &= (\hat{E}_0)_2 + \sum_{i=1}^{3} f_{2,i}((\hat{E}_0)_i) \\
&= (\hat{E}_0)_2 + f_{2,1}((\hat{E}_0)_1) + f_{2,2}((\hat{E}_0)_2) + f_{2,3}((\hat{E}_0)_3) \\
&= 0,9726 + f_{2,1}(0,7778) + f_{2,2}(0,9726) + f_{2,3}(0,7802) \\
&= 0,5963.
\end{aligned}$$

Here, the functions $f_{2,1}, f_{2,2}$ and $f_{2,3}$ are the linear functions stated above. For example $f_{2,1}((\hat{E}_0)_1)$ can be determined in the following way:

$$\begin{aligned}
f_{2,1}((\hat{E}_0)_1) &= f_{2,1}(0,7778) \\
&= a_{2,1} \times 0,7778 \\
&= -0,2018 \times 0,7778 \\
&= -0,1570.
\end{aligned}$$

Starting from the initial point $\bar{E}_0$, we calculate the next metabolic states similarly and infer the next metabolic states as in Table 6.3.

In this work, our focus is not on the stability analysis but a small discussion about stability analysis and boundedness of the solution is given in the previous chapters. As we stated in those chapters, the stability of a linear system depends on the eigenvalues of matrix M, since the Jacobian matrix J is equal to M in linear case. The eigenvalues of the matrix are determined as $\delta_1 = -0,2611, \delta_2 = -0,5139$

and $\delta_3 = -1,0547$. The eigenvalues are calculated by means of *eig* function provided by *MATLAB*. Since *all* eigenvalues are *negative*, then this implies that our system is *stable.*

6.4.2 Approximation with Polynomials

As in the previous section, we have some approximations when we infer the underlying regulatory networks by interactions represented by quadratic polynomials. For all genes $j \in \{1,2,...,n\}$ and all samples $m \in \{0,1,...,l-1\}$, the approximation changes in the following way:

$$(\dot{\bar{E}}_m)_j = (f_{j,1}((\bar{E}_m)_1) + f_{j,2}((\bar{E}_m)_2) + f_{j,3}((\bar{E}_m)_3) + ... + f_{j,n}((\bar{E}_m)_n)) + (AE_m)_j.$$

Here, $f_{j,i} : \mathbb{R} \to \mathbb{R}$ denotes the influence of the expression level of $gene_i$ to the rate of trancription of $gene_j$, being a quadratic function defined as follows: $f_{j,i}(x) = a_{j,i}x^2 + b_{j,i}x + c_{j,i}$, where $x = E_i$ denotes the concentration of $gene_i$ and $a_{j,i}, b_{j,i}, c_{j,i} \in \mathbb{R}$.

The minimization problem can in fact be separated into n subproblems, each subproblem including l approximations as follows:

$$\begin{aligned}
(\dot{\bar{E}}_0)_j &= (a_{j,1}(\bar{E}_0)_1^2 + b_{j,1}(\bar{E}_0)_1 + c_{j,1}) + ... + (a_{j,n}(\bar{E}_0)_n^2 + b_{j,n}(\bar{E}_0)_n + c_{j,n}) + (AE_0)_j,\\
(\dot{\bar{E}}_1)_j &= (a_{j,1}(\bar{E}_1)_1^2 + b_{j,1}(\bar{E}_1)_1 + c_{j,1}) + ... + (a_{j,n}(\bar{E}_1)_n^2 + b_{j,n}(\bar{E}_1)_n + c_{j,n}) + (AE_1)_j,\\
(\dot{\bar{E}}_2)_j &= (a_{j,1}(\bar{E}_2)_1^2 + b_{j,1}(\bar{E}_2)_1 + c_{j,1}) + ... + (a_{j,n}(\bar{E}_2)_n^2 + b_{j,n}(\bar{E}_2)_n + c_{j,n}) + (AE_2)_j,\\
&...\\
(\dot{\bar{E}}_{l-1})_j &= (a_{j,1}(\bar{E}_{l-1})_1^2 + b_{j,1}(\bar{E}_{l-1})_1 + c_{j,1}) + ... + (\qquad ... \qquad + c_{j,n}) + (AE_{l-1})_j.
\end{aligned}$$

For all j, the minimization of these approximation errors is done in the following simultaneous way of least squares:

$$\min_{\substack{c_{j,1},b_{j,1},a_{j,1},\\ \cdots \\ c_{j,n},b_{j,n},a_{j,n}}} \left\| \begin{bmatrix} 1 & (\bar{E}_0)_1 & ((\bar{E}_0)_1)^2 ... & 1 & (\bar{E}_0)_n & ((\bar{E}_0)_n)^2 \\ 1 & (\bar{E}_1)_1 & ((\bar{E}_1)_1)^2 ... & 1 & (\bar{E}_1)_n & ((\bar{E}_1)_n)^2 \\ 1 & (\bar{E}_2)_1 & ((\bar{E}_2)_1)^2 ... & 1 & (\bar{E}_2)_n & ((\bar{E}_2)_n)^2 \\ ... & ... & ... & ... & ... & ... \\ 1 & (\bar{E}_{l-1})_1 & ((\bar{E}_{l-1})_1)^2 ... & 1 & (\bar{E}_{l-1})_n & ((\bar{E}_{l-1})_n)^2 \end{bmatrix} \begin{bmatrix} c_{j,1} \\ b_{j,1} \\ a_{j,1} \\ c_{j,2} \\ b_{j,2} \\ a_{j,2} \\ ... \\ c_{j,n} \\ b_{j,n} \\ a_{j,n} \end{bmatrix} - \begin{bmatrix} (\dot{\bar{E}}_0)_j \\ (\dot{\bar{E}}_1)_j \\ (\dot{\bar{E}}_2)_j \\ ... \\ (\dot{\bar{E}}_{l-1})_j \end{bmatrix} \right\|^2 .$$

Thus, for all $j \in \{1, 2, ..., n\}$ the subproblem can be stated as:

$$\min_{\breve{A}_j} \sum_{m=0}^{\ell-1} (\breve{E}_m \breve{A}_j - (\dot{\bar{E}}_m)_j)^2,$$

where $\breve{E}$ is a collection of Vandermonde matrices for polynomials as represented in the optimization problem at the top of the page. Here, $\breve{E}_m$ represents the m^{th} row vector of the matrix $\breve{E}$. In addition, $\breve{A}_j = ((\breve{A}_j)_1, (\breve{A}_j)_2, (\breve{A}_j)_3 ..., (\breve{A}_j)_{3n})^T$ is a column vector consisting of the coefficients of the polynomials for the regulating functions $f_{j,1}$, $f_{j,2}$, ..., $f_{j,n}$.

Please note that dividing the problem into subproblems does not require any additional restrictions and is completely equivalent to the huge optimization problem. This special division also enables us to use parallel computing, i.e., we can compute the regulating functions' coefficients for each gene separately. Computational complexity of the system highly reduces when we use parallel computing. Another important issue to consider is that, when we do not divide the problem but instead decompose the original system, we use the sparsity of the gene interaction matrix and cause an increase in the rounding errors. What we do instead, is dividing the system into subproblems and for each problem using *Singular Value Decomposition (SVD).* More precisely, we use SVD to decompose $\breve{E}$, and analyze and improve the rank deficiency of the matrices.

Singular Value Decomposition (SVD)

The least squares problems which reveal a rank deficiency, have no unique solution [33, 36]. A common practice is to select the minimum residual solution having the smallest norm. If a least squares problem is close to rank-deficiency, then the system will be quite sensitive to the perturbations in the input data. In addition, the condition number for a matrix also measures how close a matrix is from being singular. Please note that the sensitivity to perturbations is so important for our model that we should make our system less sensitive to the perturbations in the experimental data. This is being called *regularization* or *stabilization.*

When we have ill-conditioned or rank deficient systems in least squares problems, we usually use a method of analyzing and solving the problem called *singular value decomposition (SVD).*

Let $A \in \mathbb{R}^{m \times n}$, then the singular value decomposition has the form:

$$A = USV^T,$$

where is $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices and $S \in \mathbb{R}^{m \times n}$ is a rectangular matrix which consists of a leading diagonal matrix D and 0 values around. The dimension of the matrix D is the rank of A. Thus, the diagonal entries of D, all of them being positive, are called *the singular values* of A . In addition, the columns of the orthogonal matrices correspond to *singular vectors.*

When we use SVD in solving ill-conditioned least squares problems, SVD drops the small singular values from the solution and make the system less sensitive to the perturbations. Please refer to [33, 36] for more information about SVD. *Hansen* [33] developed the *Regularization Tools Version 3.1* for analysis and solution of discrete ill-posed problems. The developed package provides different regularization strategies, enables to compare the results and draw conclusion about the strategies. The *Regularization Tools Version 3.1* is available for *MATLAB* users.

Here, we note some useful properties related to SVD and conclude what do these properties mean for our system. Let us denote the singular values of A as

$\alpha_1, \alpha_2, \alpha_3, ...\alpha_r$, where $\alpha_\mu \leqslant \alpha_{\mu+1}$ for $\mu = 1, 2, ..., r-1$.

The *condition number* of A measures closeness to rank deficiency. The rank of the matrix A is equal to the non-zero singular values and also the condition number for the matrix A, *cond(A)* can be calculated by means of singular values in the following way:

$$cond(A) = \frac{\alpha_r}{\alpha_1}.$$

Here, we comment that our regularization strategy should be decided according to our system, and the package is a useful tool in determining the best strategy. Now, we continue with our example.

Example 6.4.2.1

In Example 6.4.1.1, we analyzed the behavior of the linear approach given the experimental data in Table 6.1. Please remember that the difference quotients related to the data are calculated and shown in Table 6.2. Similar to the linear approach, we solve the optimization problem to determine the coefficients for the quadratic polynomials.

Let us firstly determine whether our system is close to rank deficiency or not using one of the functions provided by *MATLAB*, i.e., *svd.* Please note that the following matrix, $\breve{E}$ is same for each subproblem:

$$\breve{E} = \begin{bmatrix} 1 & 0,7778 & 0,6050 & 1 & 0,9726 & 0,9460 & 1 & 0,7802 & 0,6087 \\ 1 & 0,2324 & 0,0540 & 1 & 0,6999 & 0,4899 & 1 & 0,1815 & 0,0329 \\ 1 & 0,7026 & 0,4936 & 1 & 0,9657 & 0,9326 & 1 & 0,0424 & 0,0018 \\ 1 & 0,0032 & 0,0000 & 1 & 0,8392 & 0,7043 & 1 & 0,0503 & 0,0025 \\ 1 & 0,4908 & 0,2409 & 1 & 0,0975 & 0,0095 & 1 & 0,2250 & 0,0506 \end{bmatrix}.$$

Using *svd* for this matrix we determine the following singular values

$$\begin{aligned} \alpha_1 &= 0{,}0843 \\ \alpha_2 &= 0{,}5099 \\ \alpha_3 &= 0{,}8495 \\ \alpha_4 &= 1{,}0316 \\ \alpha_5 &= 4{,}6545, \end{aligned}$$

which are used to determine the *condition number* for $\breve{E}$:

$$cond(\breve{E}) = \frac{\alpha_5}{\alpha_1} = 55,2135.$$

If the condition number of the matrix was very large, then this would indicate that we have a nearly singular matrix. Please note that *svd* produces very valuable results for our least squares problem. We refer to [33, 36] for details.

For each subproblem j, we calculate the coefficients of the functions $f_{j,1}, f_{j,2}, ..., f_{j,n}$. The following polynomials represent the influence of one gene to another. Note that some of these polynomials are linear while the rest of them are quadratic. This reminds that our model produces linear functions as well as quadratic functions which constitutes an advantage for our model.

$$\begin{aligned} f_{1,1}(x) &= -1,3137x + 1,0019 \\ f_{2,1}(x) &= -2,0613x - 2,5403 \\ f_{3,1}(x) &= -0,4766x + 0,8010 \\ f_{1,2}(x) &= -1,5202x + 0,6461 \\ f_{2,2}(x) &= -9,0193x^2 + 9,7244x \\ f_{3,2}(x) &= +2,0340x^2 - 2,4042x \\ f_{1,3}(x) &= +0,3641x \\ f_{2,3}(x) &= -0,3357x \\ f_{3,3}(x) &= -0,7880x. \end{aligned}$$

Please observe that the coefficients of the functions $f_{j,1}, f_{j,2}$ and $f_{j,3}$ are calculated in the j^{th} subproblem. In other words, after solving each subproblem j, the coefficients which affect the next expression level of $gene_j$ are determined.

Similar to the linear case, we suppose the time differences between two approximations, $h_k = t_{k+1} - t_k = 1$ $(k \in \{0, 1, ..., s-2\})$, where s denotes the number of iterations made during approximation and prediction. We take the initial expression level vector $\bar{E}_0$ as the initial approximation and recursively infer the next metabolic states as seen in Table 6.4.

		$gene_1$	$gene_2$	$gene_3$			
at t_0	[	0,7778	0,9726	0,7802	]	=	$\hat{E}_0$
at t_1	[	0,2324	0,6999	0,1815	]	=	$\hat{E}_1$
at t_2	[	0,7026	0,9657	0,0424	]	=	$\hat{E}_2$
at t_3	[	0,0032	0,8392	0,0503	]	=	$\hat{E}_3$
at t_4	[	0,4908	0,0975	0,2250	]	=	$\hat{E}_4$
at t_5	[	0,9784	-0,6442	0,3997	]	=	$\hat{E}_5$

Table 6.4: *Approximation with quadratic polynomials.*

If we concentrate on Table 6.3 and Table 6.4, we can observe that our model approximates the experimental data better than the linear model. In the next section, we explain the success of the methods in detail.

The stability analysis for our nonlinear system is more complex than the linear case. In the linear case, we stated that our system is stable and also satisfies the criterion for the BIBO stability since the eigenvalues of the matrix M are all negative. In our model, we cannot state the stability or unstability for all E, but we can for some particular metabolic states, i.e., for some particular E, state the stability. The previous chapters describe the stability analysis in detail.

6.5 Comparative Results

6.5.1 Least Squares Approximation Errors

Microarray experiments, like many other experiment concerning dynamic systems, contain noise. Thus, artificially created data are more reliable for numerical comparison. Here, we use the least squares approximation errors to compare the linear and nonlinear approaches.

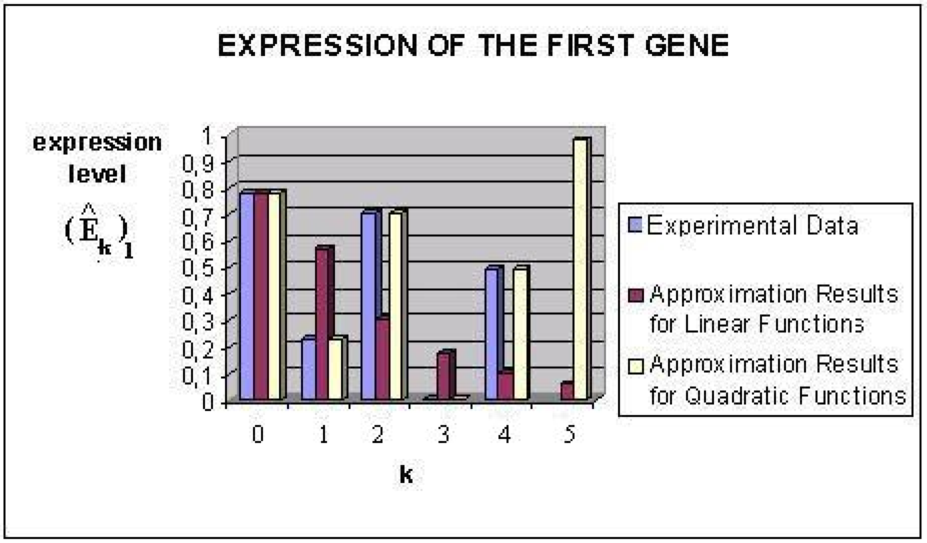


Figure 6.1: Approximated expression levels and the experimental data.

Linear approach and modeling the gene interactions with quadratic polynomials both predict the next state depending on the initial state. We note that in our examples, given in the previous section, we use the first experimental data $\bar{E}_0$ as the initial state. The success of the next predicted states depends on the model chosen. In Figure 6.1, for the example we gave in the previous sections, we analyze the approximated expression levels for the first gene.

Figure 6.1 illustrates that the approximated expression levels for the first gene are almost the same with the experimental data in the quadratic model, but in linear approximation we see that the expression levels differ from the given data. As we remember, the example data consist of five time point, i.e., $l = 4$, where the expression levels of three genes, i.e., $n = 3$, are given at each time point. In this figure, we see six time point, the last of which shows the predicted expression level for the coming state, so the last time point does not contain the experimental data, but it contains only the predicted levels. If we concentrate on the experimental data and the approximated values for the linear approach, we see that the difference between the values is not so small while this is not the case for the quadratic approach. This figure is shown only to emphasize that for only a small number of genes and discrete time intervals, the linear approach is not so successful as our model in approximation.

Please note that the expression levels form a monotone sequence, namely, decreases within time, when we use linear idealization which is an expected result because of the linearity. In the next few iterations, the expression level of the gene goes to zero which is not usually true for the real expression levels. On the other hand, because of the nonlinear behavior of our model functions, the expression level for the same gene can increase or decrease, i.e., it is not a monotone sequence, similar to the real expression levels.

A better way to compare the success of linear and nonlinear approaches is using the residuals and residual norms. Please remember that since we divided our optimization problem into n subproblems, where n denotes the number of genes being considered, we should make n comparisons. For the j^{th} optimization subproblem, we use the *residuals*, i.e., $F_j(\bar{E}_m)-(\dot{\bar{E}}_m)_j$ $(m = 0, 1, ..., l-1)$, to express the least squares error at time t_m. Thus, *residual vector* $\hat{R}_j \in \mathbb{R}^l$, $\hat{R}_j = ((\hat{R}_j)_0, (\hat{R}_j)_1, ..., (\hat{R}_j)_{l-1})^T$, for each subproblem j can then be defined in the following way:

$$\hat{R}_j = \begin{bmatrix} F_j(\bar{E}_0) - (\dot{\bar{E}}_0)_j \\ F_j(\bar{E}_1) - (\dot{\bar{E}}_1)_j \\ F_j(\bar{E}_2) - (\dot{\bar{E}}_2)_j \\ \cdots \\ F_j(\bar{E}_{l-1}) - (\dot{\bar{E}}_{l-1})_j \end{bmatrix}.$$

Table 6.6 shows the residuals for the example we gave in the previous chapters. For both linear and quadratic case, we have three columns, each standing for a subproblem, and five rows, each standing for the time point. If we want to compare the approximation errors for the first gene, i.e., the difference between the experimental data and the approximated values for the first gene, we concentrate on the first and the fourth column vectors where each entry in the column vectors correspond to the approximation error for each time point. Please observe that these column vectors correspond to the approximation errors which we observed in Figure 6.1.

Furthermore, we can use the *residual norm* for each subproblem j, which is defined as follows:

$$norm(\hat{R}_j) = \sum_{m=0}^{l-1}((\hat{R}_j)_m)^2.$$

More precisely, the residual norm is the sum of squares of the entries in the residual vector $\hat{R}_j$. The residual norm is defined as square of the minimal error value we have got by least squares approximation. Table 6.5 shows the residual norms for the same problem. Please observe that we have three rows for both linear and quadratic approaches, where each row corresponds to the subproblem.

	Linear Approach	Quadratic Approach
n	Residual Norm	Residual Norm
1	1,1780	7,1491*10^{-31}
2	0,9676	5,6091*10^{-30}
3	0,1368	3,6482*10^{-31}

Table 6.5: *Residual norms for linear and quadratic approximation (n denoting the number of genes being considered).*

	Linear Approach			Quadratic Approach		
n	1	2	3	1	2	3
	0,34	-0,10	0,08	-0,56*10^{-15}	0,09*10^{-14}	-0,56*10^{-15}
	-0,46	-0,44	0,04	-0,33*10^{-15}	0,05*10^{-14}	-0,03*10^{-15}
	0,34	-0,17	0,10	-0,22*10^{-15}	-0,19*10^{-14}	0,15*10^{-15}
	-0,30	0,61	-0,20	-0,44*10^{-15}	0,09*10^{-14}	-0,08*10^{-15}
	-0,80	0,61	-0,28	-0,22*10^{-15}	-0,01*10^{-14}	-0,17*10^{-15}

Table 6.6: *Residuals for linear and quadratic approximation (n denoting the number of genes being considered).*

These results show that nonlinear approximation generates more accurate results when we compare with the linear approach. Here, we should note that a better comparison of the approaches should be made after training the system with a large number of experimental data. After training of the system, the success of the determined parameters should be *tested* for other experimental data. More precisely, given some experimental data, we should compare the experimental data and the predicted data determined by our *time-discrete equation,* given the initial state. However, we showed earlier that our model not only produces quadratic polynomials but linear functions as well. This implies that the future behavior of the system described by a linear model can also be regarded as a special case of modeling by a quadratic polynomial.

6.5.2 Pathway Analysis

In 1998, *Spellman et al.* [66] carried a famous study for the yeast *Saccharomyces cerevisiae* and determined some of the regulator genes in the yeast. Here, we study the gene interactions between some parts of the genes which are responsible for the *glycolysis pathway.* More precisely, we study the interactions between *NTH2*, *ATH1*, *TPS1*, *TPS2*, *TSL1*, *TPS3*, *GSY2*, *GSY1* and *GLG1,* given the experimental data in Appendix.

The experimental data are public and we use these data to compare the residual norms with different models. The figures given in Appendix show the calculated values for approximation with polynomials of degree one, two and three, respectively. The results for the first, second and third degree polynomials are shown in Table 6.7.

	Linear	Quadratic	Polynomial of degree 3
n	Residual Norm	Residual Norm	Residual Norm
1	0,4714	0,0380*10^{-28}	0,1093*10^{-29}
2	0,0352	0,0389*10^{-28}	0,2790*10^{-29}
3	0,1549	0,2190*10^{-28}	0,1581*10^{-29}
4	0,1855	0,0433*10^{-28}	0,1773*10^{-29}
5	0,5443	0,0477*10^{-28}	0,3920*10^{-29}
6	0,2137	0,0732*10^{-28}	0,0579*10^{-29}
7	2,6724	0,0677*10^{-28}	0,2846*10^{-29}
8	0,5964	0,0246*10^{-28}	0,2861*10^{-29}
9	1,2069	0,3414*10^{-28}	0,4610*10^{-29}

Table 6.7: *Residual norms for approximation with first, second and third degree polynomials (n denoting the number of genes being considered).*

This table shows that the least squares errors corresponding to the linear and nonlinear polynomials are quite different. For the linear case, the relative error is quite high, i.e., the model approximates the experimental data not accurately, while quadratic polynomials can approximate the data with negligible quadratic errors. Increasing the degree of the polynomials provides a better fit to the training data. However, this results in higher sensitivity to measurement noise and process uncertainties. Therefore, polynomials with the smallest degree providing a reasonable approximation should be preferred.

Since the enzyme concentrations have multiplicative effects, there are strong nonlinearities in gene regulation. This is the most important reason for the poor performance of linear approximation. In experimental results, we obtained lower residual errors for cubic polynomials. It is natural that a *cubic* polynomial provides a better fit to a given wave-form, since quadratic polynomials are a subclass of it. However, the residual error does not give us the error but it gives the *minimum* possible error because there exist both measurement noise and process uncertainties. A perfect fit also includes all those errors as if they are part of the systems dynamics; this will result in very high prediction errors for long term. Modeling the system by a more limited range of functions is a popular method to attenuate the noise. If sufficient data were available, the correct comparison should be as follows: We use only a portion of data to train our system and estimate the parameters. Let us denote this first portion of data by $\bar{E}_0, \bar{E}_1, ..., \bar{E}_{l_1-1}$. Then, we test the model with some of the remaining data, e.g., $\bar{E}_{l_1}, \bar{E}_{l_1+1}, ..., \bar{E}_{l_1+l_2-1}$ $(l = l_1 + l_2)$. If the test ends with a reasonable approximation error, we could consider using our model to predict the later behavior. However, available data are very limited for a real test.

CHAPTER 7

GENE NETWORK ANALYSIS

7.1 Modeling of Gene Interactions

In the previous sections, we explained how to infer a gene regulatory network, given a finite set of gene expression profiles. In the literature [12, 13], the gene interactions are characterized by some functions of the expression levels of genes. In these idealizations, the modeling functions are linear, while in our approach we use quadratic polynomials. For any modeling function, we can construct a weighted directed graph corresponding to a gene regulatory network, where each gene corresponds to a vertex in the graph. The influences are represented by the directed edges and the weights of the influences are described by the modeling functions.

In our model, we do *not* have any restriction on the regulatory dynamics. For example, there is no restriction on the number of regulator genes for a gene that corresponds to the input edges targeting that edge. Furthermore, we also allow self-regulating genes which correspond to the loops in the network. The followed strategy constructs a gene regulatory network, where each gene in the network can both have influences on some genes and be influenced by some other genes including itself in the network.

Here, we give a brief summary of gene networks. We first assume that the gene interactions are *constant*, i.e., the influence of any gene on another gene does not change. For such a case, the next influence of the genes does not change whatever the expression level of the influencing gene is. This special case corresponds to a static network, where each weight is constant. Please note that such a modeling cannot give an inference for the underlying dynamics in reality, since the underlying network topology is fixed for such a definition.

When we define the weights as *linear* functions, we have an increased dynamics in the network. The influence of any gene to another gene changes linearly, depending on the expression level of the regulating gene. We discussed the linear approaches [12, 13, 81] in Chapter 3. In our approach, we assume *nonlinear* interactions between genes. In the following sections, we consider a particular nonlinear function, namely, *quadratic* polynomials.

7.1.1 Constant Influences

Assume that the genes that have an influence on the expression level of $gene_j$ ($j \in \{1, 2, ..., n\}$) are represented by a set $I_j = \{gene_{j,i_1^j}, gene_{j,i_2^j}, ..., gene_{j,i_{n_j}^j}\}$, where $i_1^j, ..., i_{n_j}^j \in \mathbb{N}$ and $n_j \leqslant n \in \mathbb{N}$. Furthermore, the influence levels are given by the set $\left\{a_{j,i_1^j}, a_{j,i_2^j}, ..., a_{j,i_{n_j}^j}\right\}$ with $a_{j,i_\kappa^j} \in \mathbb{R}$ for $\kappa \in \{1, 2, ..., n_j\}$. In other words, assuming that the influence of $gene_i$ *to* $gene_j$ is described by a constant function $f_{j,i} \equiv a_{j,i}$, for the next state, the change in the expression level of $gene_j$ will be a constant value, i.e.,

$$F_j(E) = \sum_{\kappa=1}^{n_j} f_{j,i_\kappa^j}(E_{i_\kappa^j}) = \sum_{\kappa=1}^{n_j} a_{j,i_\kappa^j},$$

whatever the expression levels of influencing genes, i.e., $E_{i_1^j}, E_{i_2^j}, ..., E_{i_{n_j}^j}$, are.

Based on the definitions in [9, 85], we formally define such *a weighted directed network* in the following way: A weighted directed network is an ordered triple $G = (V, E, W)$, where $V = \{v_i | i = 1, 2, 3, ..., n\}$ is a finite, non-empty set of $n \in \mathbb{N}$ different elements, where each element in the set is called a *vertex*. Furthermore, E represents a finite set of ordered pairs of vertices, i.e., $E \subseteq \{e_{i,j} = (v_i, v_j) | v_i, v_j \in V\}$, and $W = \{(e_{i,j}, w_{j,i}) | e_{i,j} \in E\}$ is a relation set which associates an edge with a weight value, where $(f_{j,i} \equiv)\ w_{j,i} \in \mathbb{R}$.

By our introduction of a weighted directed network, we allow that there are *loops*, i.e., edges starting and terminating in the same vertex, and that an edge may go in both directions between two different vertices. Figure 7.1 shows a gene network of three genes, where the gene expression levels also shown on the left-hand side. We

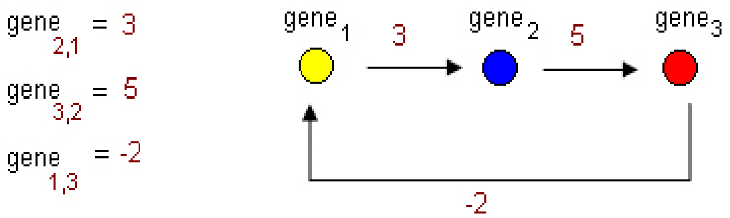


Figure 7.1: A directed graph with constant weights for a gene network consisting of three genes whose influences are shown on the left-hand side.

formally define this network by these sets:

$$\begin{aligned} V &= \{v_i | i = 1, 2, 3\}, \\ E &= \{e_{1,2} = (v_1, v_2), e_{2,3} = (v_2, v_3), e_{3,1} = (v_3, v_1)\}, \\ W &= \{(e_{1,2}, 3), (e_{2,3}, 5), (e_{3,1}, -2)\}. \end{aligned}$$

Please note that for this problem we have $w_{2,1} = 3, w_{3,2} = 5, w_{1,3} = -2$.

7.1.2 Linear Idealization

In Section 3.3, we discussed the linear approach and represented the influence of $gene_i$ *to* $gene_j$ by a function $f_{j,i}(x) = a_{j,i}x$, where $a_{j,i} \in \mathbb{R}$ and $x = E_i$ denotes the expression level of $gene_i$. The definition given for a static network in Section 7.1.1 remains the same for the network definition which we use now for linear idealization, except the relation set. Here, $W = \{(e_{i,j}, w_{j,i}) | e_{i,j} \in E\}$ represents the relation between edges and influence functions where $w_{j,i} = f_{j,i}(x) = a_{j,i}x$ is an influence function of $x = E_i \in \mathbb{R}$, denoting the expression level of the i^{th} gene.

Assume that the genes which have an influence on the expression level of $gene_j$ are represented by a set $I_j = \{gene_{j,i_1^j}, gene_{j,i_2^j}, ..., gene_{j,i_{n_j}^j}\}$, similar to the case we consider the constant influences. The change in the expression level of $gene_j$ is modelled by the sum of products of expression levels of the influencing genes, i.e., $E_{i_1^j}, E_{i_2^j}, ..., E_{i_{n_j}^j}$, and the influence levels, $a_{j,i_1^j}, a_{j,i_2^j}, ..., a_{j,i_{n_j}^j}$. More precisely, the

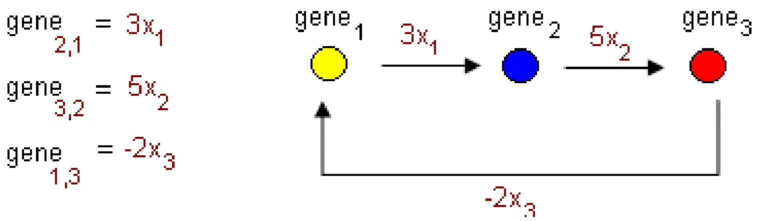


Figure 7.2: A directed graph with linear weights for a gene network consisting of three genes whose influence functions are shown on the left-hand side.

change in the expression level of $gene_j$ is

$$F_j(E) = \sum_{\kappa=1}^{n_j} f_{j,i^j_\kappa}(E_{i^j_\kappa}) = \sum_{\kappa=1}^{n_j} a_{j,i^j_\kappa} E_{i^j_\kappa}.$$

Figure 7.2 shows a gene network of three genes, where the gene influence functions are also shown on the left-hand side. In this figure, x_1, x_2 and x_3 denote the expression levels of the first, second and third genes, i.e., E_1, E_2 and E_3, respectively. We formally define this network by these sets:

$$\begin{aligned} V &= \{v_i | i = 1, 2, 3\}\,, \\ E &= \{e_{1,2} = (v_1, v_2), e_{2,3} = (v_2, v_3), e_{3,1} = (v_3, v_1)\}\,, \\ W &= \{(e_{1,2}, w_{2,1}), (e_{2,3}, w_{3,2}), (e_{3,1}, w_{1,3})\}, \end{aligned}$$

where $w_{2,1} = 3x$ $(x = E_1)$, $w_{3,2} = 5x$ $(x = E_2)$ and $w_{1,3} = -2x$ $(x = E_3)$.

From our representation of the change in the expression level of $gene_j$ by F_j, i.e., by the right-hand side of the system dynamics $\dot{E} = F(E)$, we learn the close relation between our system of ODEs and the corresponding gene network. While here we have referred to the linear case $F(E) = ME$, in the following we shall refer to our quadratic case of modeling.

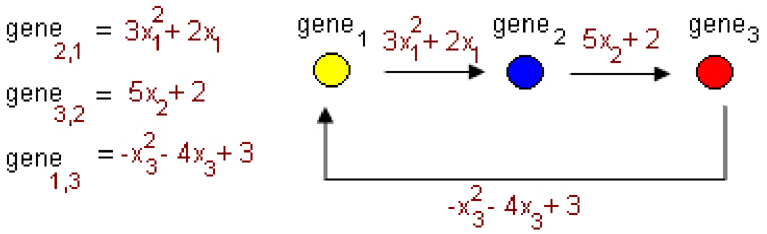


Figure 7.3: A directed graph whose weights are given by quadratic polynomials on the left-hand side, corresponding to a gene network of three genes.

7.1.3 Modeling with Quadratic Polynomials

In our approach, we represent the interactions by quadratic polynomials. As we did in the previous chapters, now we represent the influence of $gene_i$ *to* $gene_j$ by a quadratic function $w_{j,i} = f_{j,i}(x) = a_{j,i}x^2 + b_{j,i}x + c_{j,i}$, where $a_{j,i}, b_{j,i}, c_{j,i} \in \mathbb{R}$ and $x = E_i$ denotes the expression level of $gene_i$.

Assume that all genes which have an influence on the expression level of $gene_i$, are represented by a set $I_j = \{gene_{j,i^j_1}, gene_{j,i^j_2}, ..., gene_{j,i^j_{n_j}}\}$. The change in the expression level of $gene_i$ is approximated by the corresponding influence functions. More precisely, the change in the expression level of $gene_j$ is

$$F_j(E) = \sum_{\kappa=1}^{n_j} f_{j,i^j_\kappa}(E_{i^j_\kappa}) = \sum_{\kappa=1}^{n_j}(a_{j,i^j_\kappa}(E_{i^j_\kappa})^2 + b_{j,i^j_\kappa}E_{i^j_\kappa} + c_{j,i^j_\kappa}).$$

Figure 7.3 shows a gene network of three genes, where the gene influence polynomials are also shown on the left-hand side. We formally define this network by these sets:

$$\begin{aligned} V &= \{v_i | i = 1, 2, 3\}\,, \\ E &= \{e_{1,2} = (v_1, v_2), e_{2,3} = (v_2, v_3), e_{3,1} = (v_3, v_1)\}\,, \\ W &= \{(e_{1,2}, w_{2,1}), (e_{2,3}, w_{3,2}), (e_{3,1}, w_{1,3})\}. \end{aligned}$$

Here, $w_{2,1} = 3x^2 + 2x$ $(x = E_1)$, $w_{3,2} = 5x + 2$ $(x = E_2)$ and $w_{1,3} = -x^2 - 4x + 3$ $(x = E_3)$.

7.2 Analysis of Gene Networks

Given a gene regulatory network, we can analyze the underlying topology depending on the biological questions to be answered. If we are given a network where the vertex set consists of genes in a particular metabolic reaction and we are asked to find the effects of lack of any gene in this reaction, we search for cut vertices in the given network and analyze the network *connectivity* when any *fixed* cut vertex is removed from the network. This is also related with so-called *knockout* experiments. If the question concerns the influence of a *fixed* gene to others, we analyze the shortest paths from the *fixed* vertex to others. Similarly, we compute the shortest path from all vertices to one *fixed* vertex to express the influence of all genes to the *fixed* gene. Please refer to [16] for the graph theoretical foundations and algorithms.

With current technology and biological information in hand, we analyze particular groups of genes, for example, the interactions between genes functioning in consecutive metabolic reactions. Analyzing larger networks, for example, analyzing human genome which consists of thousands of genes interacting with each other, needs other questions to be answered. In addition to the questions asked for small networks, such as finding cut vertices or a shortest path between two distinct vertices, some other questions arise, e.g., how the network connectivity is affected by deleting some percentage of vertices, or what is the average shortest path length for current larger networks.

The coming sections concern the main problems to be answered in the analysis of gene networks.

7.2.1 Degree Related Questions

To indicate the number of genes which have an influence on a particular gene, we use the term *indegree*. The indegree of a vertex, say v_i with $i \in \{1, 2, ..., n\}$, denotes the number of directed edges whose target is that particular vertex. We denote the indegree of vertex v_i by $in(v_i)$. Similarly, when we want to know the number of genes which are influenced by a particular gene, we speak about the *outdegree* of that vertex which is defined as the number of edges originating from the target vertex, $out(v_i)$.

Indegree and outdegree of a vertex are the cardinalities of the following edge sets:

$$\begin{aligned} A(v) = & \ \{(v_i, v_j) \in E | v_j = v\}, \\ B(v) = & \ \{(v_i, v_j) \in E | v_i = v\}. \end{aligned}$$

7.2.2 Path Related Questions

Biologists usually ask questions about the relation between some genes, where the question can be answered by means of *paths*. A path in a directed weighted graph $G = (V, E, W)$ is a non-empty sequence of vertices

$$P = (v_1, v_2, ..., v_k),$$

where $v_i \in V$ $(i \in \{1, 2, ..., n\})$ such that $(v_i, v_{i+1}) \in E$ $(i \leqslant n - 1)$. The *length* of the path is $k - 1$. If there is a path connecting one vertex to a particular vertex, then we can say that the gene has an influence on that particular gene.

By means of paths we detect whether there is one gene regulating another, but the answer is not quantitative. To answer such questions, we use *shortest path algorithms* to determine the weak and strong interactions between genes. The shortest path problem is discussed in the coming sections. Another interesting and for our research promising approach to the connectedness in gene networks is given by the properties of *gossiping* and *broadcasting*. Both of them have already successfully been studied for special network classes [22].

7.2.3 Shortest Path Problem

Gene network analysis mainly concerns shortest path problems, which give a strong clue about weak and strong interactions between genes. In a static network, finding shortest paths from a vertex can be solved by *Dijkstra's Algorithm* [17]. This algorithm gives us the enumerated shortest paths from a source vertex to other vertices. This shows the influence of the source gene to other genes. Similarly, if the shortest path from one vertex to a destination vertex is smaller than the shortest path from another vertex to the destination, we conclude that the influence of the second gene is stronger that the influence of the first one.

An important note should be stated here: *Dijkstra's Algorithm* is used for directed, acyclic networks. A *directed, acyclic graph* is a directed graph which contains no *cycles*. The examples we have given in the previous sections contain *cycles*, i.e., there are paths whose start vertices and end vertices are the same. We cannot use *Dijkstra's Algorithm* for finding shortest paths for such examples. This constitutes a challenge for the nature of the regulations, since feedback loops in the regulatory relations generate a cycle in the gene regulatory network.

7.3 Choice of Representation and Algorithm

For the shortest path problems, there are some other algorithms with better complexity than *Dijkstra's Algorithm* mentioned in the literature [9, 14, 60]. Here, we only give a motivation to the problem of weak and strong interactions and provide an introduction to graph theory. Please refer to [16, 23] for closer details on graph theory, and [46, 67, 83] for more information on the algorithms and data structures used for networks.

Here, we should note that the best choice among all shortest path algorithms depends on the underlying biological motivation. In order to make the appropriate choice when selecting a graph representation scheme and algorithm, it is necessary to understand the time and space issues. Space and time considerations depend on the current network and underlying topology.

There is always a *trade-off* between the complexity of an algorithm and the space allocation for the data structures. So, given a gene interaction matrix, one should decide on an optimum decision. We contribute that depending on the biological motivation and biological questions to be answered, we should decide on the underlying data structure and choose a suitable algorithm for analyzing the weak and strong interactions between genes.

CHAPTER 8

SUMMARY AND CONCLUSION

In this study, we focussed on the inverse problem of *inference* of the gene regulatory dynamics based on time-series gene expression patterns. Our particular interest was concentration on the model class selection. Here, we underlined that the model should describe the continuous nature of the biochemical processes and reflect the nonlinearities when approximating and predicting the gene expression patterns. We used the gene expression rates by *ordinary differential equations (ODEs)* where each regulator effect is modeled by a nonlinear function, in particular, a quadratic polynomial. The expression rates were approximated by means of difference quotients.

In fact, our problem is an inverse problem which aims to find the polynomial coefficients that best fit the training data. The coefficients of these polynomials were determined by least squares approximation. We studied the corresponding general cases of *solvability*, assumed that the experimental data are sufficient not to have a rank deficient system, and clarified the selection of the model class of *quadratic polynomials* for gene interactions.

Furthermore, we also considered the *stability* of the referred dynamical system. We discussed that a linear system is stable if it is stable for any particular state (especially, any stationary point), while biological systems which consist of highly nonlinear mechanisms may be stable for some range and unstable for another range. More precisely, we explained that when we have a linear system, the gene interaction matrix is a constant matrix. Thus, if the matrix satisfies the stability criterion for any expression vector $E = E(t)$, at any time t, then the system is stable, since the interaction matrix is constant. On the other hand, when we have a nonlinear system, we explained that we cannot easily state the stability of our system, but we can analyze the stability of the system by assigning suitable values to the model parameters. In other words, we can divide the subspace spanned by the coefficients to separate the stable and unstable ranges. We also discussed the BIBO stability which is a useful criterion to generate an accurate dynamical system and explained that BIBO stability studies the response of the system to a bounded input, i.e., perturbation, more precisely the boundedness of our system.

The *time-discrete equation*, which we carefully provided and discussed, is an autonomous system, i.e., the system does not depend on time but it depends only on the current state. The anticipation of the gene expressions is described by the initial value provided to the system and the recursive definition for the next states. So, the accuracy of the predicted values depends on the initial value given to the system.

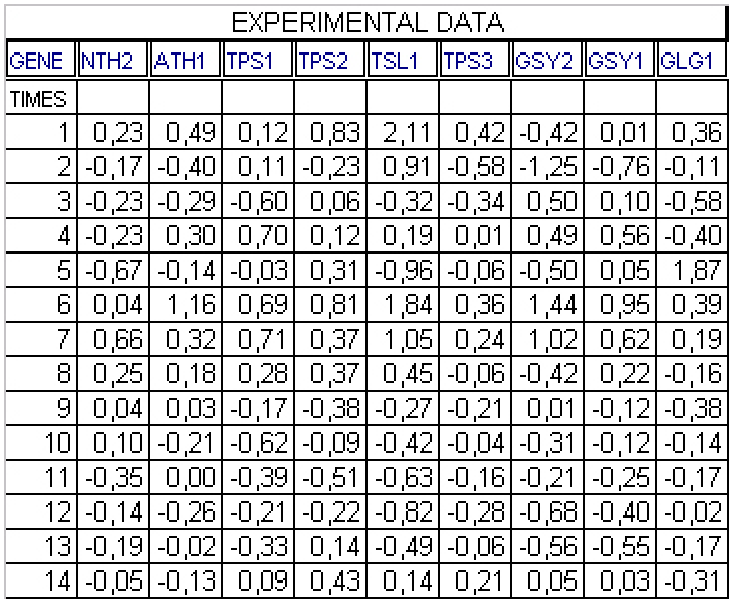
Estimated parameters are used to construct a gene regulatory network. We discussed the gene network analysis where we stated that for nonlinear systems, determining the quantitative regulatory effects is not so simple. Even for a quadratic polynomial and very few edges, we have to decide on the shortest paths which are also optimization problems. The difficulty of analyzing such a complex and continuously depending network constitutes a big challenge for determining valuable results. However, we have an opportunity to show the Boolean relations, e.g., a particular gene has an effect or no effect on some other gene. We concluded that the data structures and algorithms to be used in the analysis of a particular biosystem should be chosen depending on the biological questions to be answered and on the underlying biological motivation.

Modeling and anticipation of the gene expressions and inference of the gene regulatory networks is one of the most interesting problems in dynamical systems. Highly nonlinear mechanisms constitute a challenge for the anticipation of the gene expressions. Another challenge consists in compensating the insufficiency about the biochemical reactions, enzyme kinetics, chemical affinities, binding factors and unknown variables, which all have effects on the regulatory dynamics.

Describing the nonlinear structure of the gene regulations is a pioneering approach and an improvement for the approach of linear idealization. In this study, we particularly worked with quadratic polynomials. For future work, we recommend to work on polynomials, splines, trigonometric, exponential functions and solvable combinations of these functions to represent the dynamics of the regulatory relations.

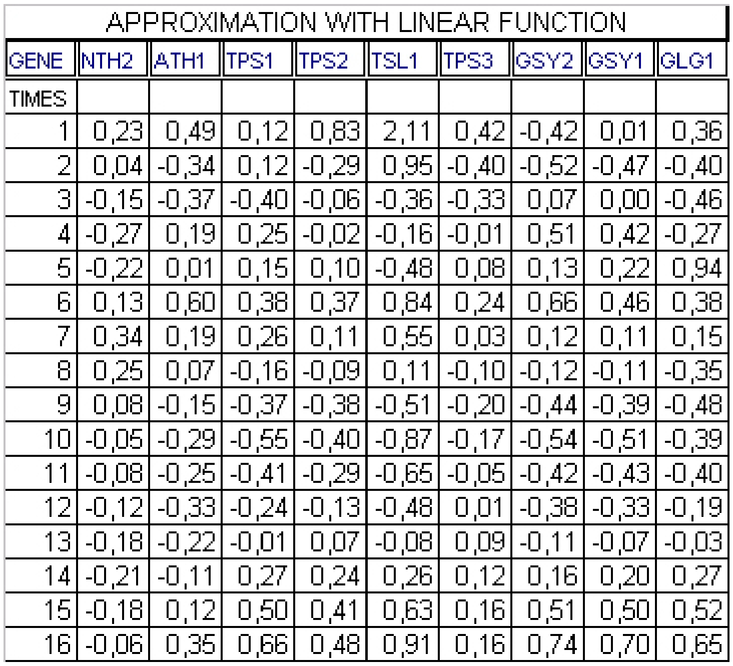
Finally, with teachers and students from Institute of Applied Mathematics of Middle East Technical University and international colleagues, *anticipatory (predictive) systems* are under investigation [21]. For these systems, where our study of gene patterns in time and its future extensions are a central motivation for, generalized semi-infinite optimization [68, 82] is a further promising mathematical technology.

APPENDIX



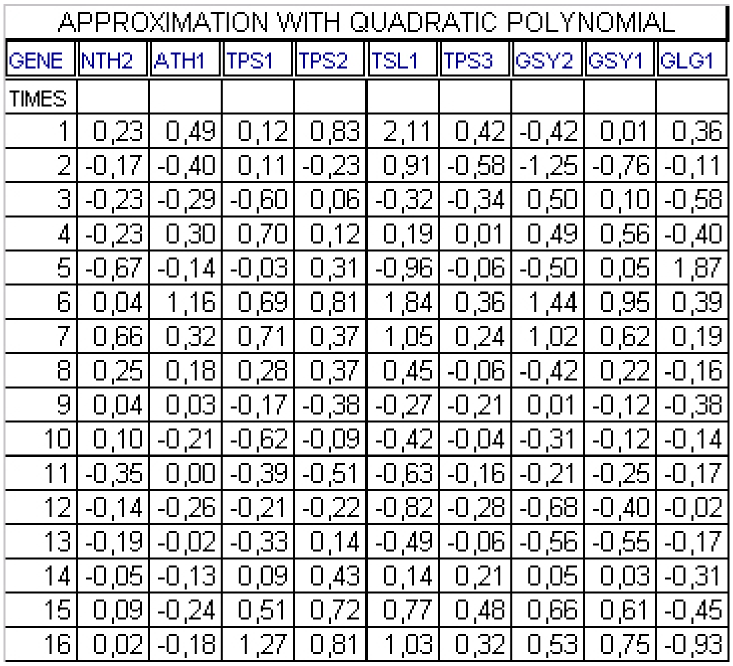
EXPERIMENTAL DATA									
GENE	NTH2	ATH1	TPS1	TPS2	TSL1	TPS3	GSY2	GSY1	GLG1
TIMES									
1	0,23	0,49	0,12	0,83	2,11	0,42	-0,42	0,01	0,36
2	-0,17	-0,40	0,11	-0,23	0,91	-0,58	-1,25	-0,76	-0,11
3	-0,23	-0,29	-0,60	0,06	-0,32	-0,34	0,50	0,10	-0,58
4	-0,23	0,30	0,70	0,12	0,19	0,01	0,49	0,56	-0,40
5	-0,67	-0,14	-0,03	0,31	-0,96	-0,06	-0,50	0,05	1,87
6	0,04	1,16	0,69	0,81	1,84	0,36	1,44	0,95	0,39
7	0,66	0,32	0,71	0,37	1,05	0,24	1,02	0,62	0,19
8	0,25	0,18	0,28	0,37	0,45	-0,06	-0,42	0,22	-0,16
9	0,04	0,03	-0,17	-0,38	-0,27	-0,21	0,01	-0,12	-0,38
10	0,10	-0,21	-0,62	-0,09	-0,42	-0,04	-0,31	-0,12	-0,14
11	-0,35	0,00	-0,39	-0,51	-0,63	-0,16	-0,21	-0,25	-0,17
12	-0,14	-0,26	-0,21	-0,22	-0,82	-0,28	-0,68	-0,40	-0,02
13	-0,19	-0,02	-0,33	0,14	-0,49	-0,06	-0,56	-0,55	-0,17
14	-0,05	-0,13	0,09	0,43	0,14	0,21	0,05	0,03	-0,31

Figure A1: Time-series data for the genes in the glycolysis pathway [66].



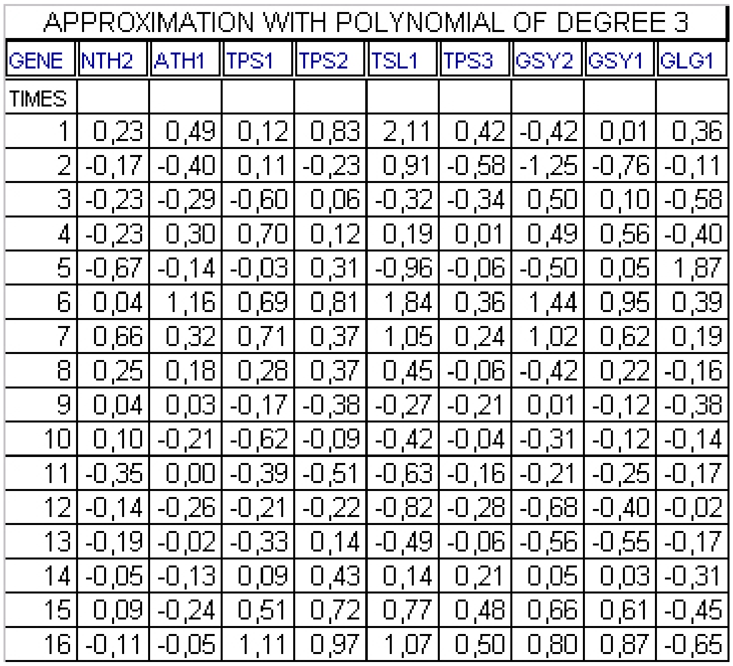
APPROXIMATION WITH LINEAR FUNCTION									
GENE	NTH2	ATH1	TPS1	TPS2	TSL1	TPS3	GSY2	GSY1	GLG1
TIMES									
1	0,23	0,49	0,12	0,83	2,11	0,42	-0,42	0,01	0,36
2	0,04	-0,34	0,12	-0,29	0,95	-0,40	-0,52	-0,47	-0,40
3	-0,15	-0,37	-0,40	-0,06	-0,36	-0,33	0,07	0,00	-0,46
4	-0,27	0,19	0,25	-0,02	-0,16	-0,01	0,51	0,42	-0,27
5	-0,22	0,01	0,15	0,10	-0,48	0,08	0,13	0,22	0,94
6	0,13	0,60	0,38	0,37	0,84	0,24	0,66	0,46	0,38
7	0,34	0,19	0,26	0,11	0,55	0,03	0,12	0,11	0,15
8	0,25	0,07	-0,16	-0,09	0,11	-0,10	-0,12	-0,11	-0,35
9	0,08	-0,15	-0,37	-0,38	-0,51	-0,20	-0,44	-0,39	-0,48
10	-0,05	-0,29	-0,55	-0,40	-0,87	-0,17	-0,54	-0,51	-0,39
11	-0,08	-0,25	-0,41	-0,29	-0,65	-0,05	-0,42	-0,43	-0,40
12	-0,12	-0,33	-0,24	-0,13	-0,48	0,01	-0,38	-0,33	-0,19
13	-0,18	-0,22	-0,01	0,07	-0,08	0,09	-0,11	-0,07	-0,03
14	-0,21	-0,11	0,27	0,24	0,26	0,12	0,16	0,20	0,27
15	-0,18	0,12	0,50	0,41	0,63	0,16	0,51	0,50	0,52
16	-0,06	0,35	0,66	0,48	0,91	0,16	0,74	0,70	0,65

Figure A2: Approximation and inference by linear functions.



APPROXIMATION WITH QUADRATIC POLYNOMIAL									
GENE	NTH2	ATH1	TPS1	TPS2	TSL1	TPS3	GSY2	GSY1	GLG1
TIMES									
1	0,23	0,49	0,12	0,83	2,11	0,42	-0,42	0,01	0,36
2	-0,17	-0,40	0,11	-0,23	0,91	-0,58	-1,25	-0,76	-0,11
3	-0,23	-0,29	-0,60	0,06	-0,32	-0,34	0,50	0,10	-0,58
4	-0,23	0,30	0,70	0,12	0,19	0,01	0,49	0,56	-0,40
5	-0,67	-0,14	-0,03	0,31	-0,96	-0,06	-0,50	0,05	1,87
6	0,04	1,16	0,69	0,81	1,84	0,36	1,44	0,95	0,39
7	0,66	0,32	0,71	0,37	1,05	0,24	1,02	0,62	0,19
8	0,25	0,18	0,28	0,37	0,45	-0,06	-0,42	0,22	-0,16
9	0,04	0,03	-0,17	-0,38	-0,27	-0,21	0,01	-0,12	-0,38
10	0,10	-0,21	-0,62	-0,09	-0,42	-0,04	-0,31	-0,12	-0,14
11	-0,35	0,00	-0,39	-0,51	-0,63	-0,16	-0,21	-0,25	-0,17
12	-0,14	-0,26	-0,21	-0,22	-0,82	-0,28	-0,68	-0,40	-0,02
13	-0,19	-0,02	-0,33	0,14	-0,49	-0,06	-0,56	-0,55	-0,17
14	-0,05	-0,13	0,09	0,43	0,14	0,21	0,05	0,03	-0,31
15	0,09	-0,24	0,51	0,72	0,77	0,48	0,66	0,61	-0,45
16	0,02	-0,18	1,27	0,81	1,03	0,32	0,53	0,75	-0,93

Figure A3: Approximation and inference by quadratic polynomials.



APPROXIMATION WITH POLYNOMIAL OF DEGREE 3									
GENE	NTH2	ATH1	TPS1	TPS2	TSL1	TPS3	GSY2	GSY1	GLG1
TIMES									
1	0,23	0,49	0,12	0,83	2,11	0,42	-0,42	0,01	0,36
2	-0,17	-0,40	0,11	-0,23	0,91	-0,58	-1,25	-0,76	-0,11
3	-0,23	-0,29	-0,60	0,06	-0,32	-0,34	0,50	0,10	-0,58
4	-0,23	0,30	0,70	0,12	0,19	0,01	0,49	0,56	-0,40
5	-0,67	-0,14	-0,03	0,31	-0,96	-0,06	-0,50	0,05	1,87
6	0,04	1,16	0,69	0,81	1,84	0,36	1,44	0,95	0,39
7	0,66	0,32	0,71	0,37	1,05	0,24	1,02	0,62	0,19
8	0,25	0,18	0,28	0,37	0,45	-0,06	-0,42	0,22	-0,16
9	0,04	0,03	-0,17	-0,38	-0,27	-0,21	0,01	-0,12	-0,38
10	0,10	-0,21	-0,62	-0,09	-0,42	-0,04	-0,31	-0,12	-0,14
11	-0,35	0,00	-0,39	-0,51	-0,63	-0,16	-0,21	-0,25	-0,17
12	-0,14	-0,26	-0,21	-0,22	-0,82	-0,28	-0,68	-0,40	-0,02
13	-0,19	-0,02	-0,33	0,14	-0,49	-0,06	-0,56	-0,55	-0,17
14	-0,05	-0,13	0,09	0,43	0,14	0,21	0,05	0,03	-0,31
15	0,09	-0,24	0,51	0,72	0,77	0,48	0,66	0,61	-0,45
16	-0,11	-0,05	1,11	0,97	1,07	0,50	0,80	0,87	-0,65

Figure A4: Approximation and inference by polynomials of degree three.

REFERENCES

[1] Akhmet, M. U, Gebert, J., Öktem, H., Pickl, S. W. and Weber, G.-W. (2003) *An improved method for analytical modeling and anticipation of gene expression patterns*, Preprint, Middle East Technical University, Institute of Applied Mathematics.

[2] Akutsu, T., Miyano, S. and Kuhara, S. (1999) *Identification of genetic networks from a small number of gene expression patterns under the boolean network model*, Pacific Symposium on Biocomputing 4, pp:17-28.

[3] Alligood, K. T., Sauer, T. D. and Yorke, J. A. (1997) *Chaos: An Introduction to Dynamical Systems*, Springer-Verlag.

[4] Amann, H. (1983) *Gewöhnliche Differentialgleichungen*, Walter de Gruyter, Berlin, NewYork.

[5] Aster, R. C., Borchers, B. and Thurber, C. H. (2004) *Parameter Estimation and Inverse Problems*, Academic Press.

[6] Avery, O. T., MacLeod, C. M. and McCarty, M. (1995) *Studies on the chemical nature of the substance inducing transformation of pneumococcal types. Induction of transformation by a deoxyribonucleic acid fraction isolated from pneumococcus type III*, Mol Med.79, pp:137-157.

[7] Baldi, P. and Hatfield, G. W. (2002) *DNA Microarrays and Gene Expression: From Experiments to Data Analysis and Modeling*, Cambridge University Press.

[8] Bartle, R. G. (1975) *The Elements of Real Analysis*, Second Edition, John Wiley & Sons.

[9] Buriol, L. S., Resende, M. G. C. and Thorup, M. (2003) *Speeding up dynamic shortest path algorithms*, submitted to Informs J. on Computing.

[10] Brayton, R. K. and Tong, C. H. (1979) *Stability of dynamical systems: A constructive approach*, IEEE Transactions on Circuits and Systems, CAS-26, pp:224-234.

[11] Brutlag, D. L., Galper, A. R., and Millis, D. H. (1991) *Knowledge-based simulation of DNA metabolism: Prediction of enzyme action*, Comput. Appl. Biosci. 7, 1, pp:9-19.

[12] Chen, T. and He, H. L. (1999) *Modeling gene expression with differential equations*, Pacific Symposium on Biocomputing 4, pp:29-40.

[13] de Hoon, M. J. L., Imoto, S. and Miyano, S. (2002) *Inferring gene regulatory networks from time-ordered gene expression data using differential equations*, Proceedings of the 5th International Conference on Discovery Sciences, Lecture Note in Artificial Intelligence, 2534, pp:267-274, Springer-Verlag.

[14] Demetrescu, C., Frigioni, D., Spaccamela, A. M. and Nanni, U. (2000) *Maintaining shortest paths in digraphs with arbitrary arc weights: An experimental study*, Proceedings of the 3rd Workshop on Algorithm Engineering, Lecture Notes on Computer Science 1982, pp:218-229.

[15] DeRisi, J. L., Iyer, V. R. and Brown, P.O. (1997) *Exploring the metabolic and genetic control of gene expression on a genomic scale,* Science 278, pp:680-686.

[16] Diestel, R. (1997) *Graph Theory*, New York, Springer.

[17] Dijkstra, E. W. (1959) *A note on two problems in connection with graphs*, Numerical Math. 1, pp:269-271.

[18] Dutilh, B. (1999) *Analysis of data from microarray experiments, the state of the art in gene network reconstruction*, Literature thesis, Utrecht University, Utrecht, Nederlands.

[19] Edwards, R. and Glass, L. (2000) *Combinatorial explosion in model gene networks,* Chaos 10, pp:691-704.

[20] Erb, R. S. and Michaels, G. S. (1999) *Sensitivity of biological models to errors in parameter estimates*, Pacific Symposium on Biocomputing 4, pp:53-64.

[21] Ergenç, T., Pickl, S.W., Radde, N. and Weber, G.-W. (2004) *On the topology of generalized semi-infinite optimization and anticipatory systems*, to appear in International Journal Computing Anticipatory Systems, 14-16.

[22] Fertin, G. and Raspaud, A. (2004) *A survey on Knödel graphs*, Discrete Applied Mathematics 137, pp:173-195.

[23] Frank, H. (1969) *Graph Theory*, Reading, Mass., Addison-Wesley Pub. Co.

[24] Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000) *Using Bayesian networks to analyze expression data*, J. Comput. Biol. 7, 3-4, pp:601-20.

[25] Gebert, J., Lätsch, M., Pickl, S.W., Weber, G.-W. and Wünschiers, R. (2004) *Genetic networks and anticipation of gene expression patterns*, Computing Anticipatory Systems: CASYS'03 - Sixth International Conference, AIP Conference Proceedings 718, pp:474-485.

[26] Gebert, J., Lätsch, M., Quek, E.M.P. and Weber, G.-W. (2004) *Analyzing and optimizing genetic network structure via path-finding*, Journal of Computational Technologies 9, 3, pp:3-12.

[27] Gebert, J., Öktem, H., Pickl, S. W., Radde, N., Weber, G.-W. and Yılmaz, F. B. (2004) *Inference of gene expression patterns by using a hybrid system formulation - an algorithmic approach to local state transition matrices*, will appear in: Proceedings IIAS (International Institute for Advanced Studies), 16th International Symposium for Anticipatory and Predictive Systems, Baden Baden.

[28] Gebert, J., Lätsch, M., Pickl, S. W., Weber, G.-W. and Wünschiers, R. (2004) *An algorithm to analyze stability of gene-expression pattern*, will appear in the special issue *Discrete Mathematics and Data Mining* of Discrete Applied Mathematics, guest editors: Boros, E., Hammer, P.L. and Kogan, A..

[29] Gebert, J. and Radde, N. (2004) *Modelling gene regulatory networks with piecewise linear differential equations*, preprint, Center of Applied Computer Science, University of Cologne, and talks held at EURO Summer Institute ”Optimization and Data Mining”, Ankara, Turkey.

[30] Galper, A. R., Brutlag, D. L., and Millis, D. H. (1993) *Knowledge-based simulation of DNA metabolism: Prediction of action and envisionment of pathways*, in: L. Hunter, ed., Artificial Intelligence and Molecular Biology, pp:365-395, AAAI Press/MIT Press, Menlo Park, CA/Cambridge, MA.

[31] Glass, L. and Kauffman, S. A. (1973) *The logical analysis of continuous, nonlinear biochemical control networks*, J. Theor. Biol. 39, pp:103-129.

[32] Guyton, A. C. and Hall, J. E. (2001) *Textbook of Medical Physiology*, 10th edition, Rittenhouse Book Distributors.

[33] Hansen, P.C. (1994) *Regularization Tools: A Matlab package for analysis and solution of discrete ill-posed problems*, Numerical Algorithms 6, pp:1-35.

[34] Hartemink, A. J., Gifford, D. K., Jaakkola, T. S. and Young, R. A. (2001) *Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks*, Pacific Symposium on Biocomputing 01.

[35] Hastie, T., Tibshirani, R. and Friedman, J. (2001) *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, Springer.

[36] Heath, M.T (2002) *Scientific Computing: An Introductory Survey*, McGraw-Hill.

[37] Hertz J. (1998) *Statistical issues in reverse engineering of genetic networks*, Pacific Symposium on Biocomputing 98.

[38] Heylighen, F. (2000) *Evolutionary transitions: how do levels of complexity emerge?*, Complexity 6, 1.

[39] Heylighen, F., Joslyn, C. and Turchin, V. (1995) *The quantum of evolution*, World Futures, the Journal of General Evolution 45, 1-4.

[40] Hishiki, T., Collier, N., Nobata, C., Ohta, T.O., Ogata, N., Park, H., Nobota, C., Sekimizu, T. and Tsujii, J. (1998) *Developing NLP Tools for Genome Informatics: An Information Extraction Perspective*, Genome Informatics 9, pp:81-90.

[41] Imoto, S., Goto, T. and Miyano, S. (2002) *Estimation of genetic networks and functional structures between genes by using bayesian networks and nonparametric regression*, Pacific Symposium on Biocomputing 02, pp:175-86.

[42] Isaacson, E. and Keller, H. B. (1966) *Analysis of Numerical Methods*, John Wiley & Sons.

[43] Iserles, A. and Crighton, D. G. (editors) *A First Course in the Numerical Analysis of Differential Equations*, Cambridge University Press.

[44] Jong, H. D. (2002) *Modeling and simulation of genetic regulatory systems: A literature review*, Computational Biology 9, 1, pp:67-103.

[45] Klug, W. S. and Cummings, M. R. (2003) *Concepts of Genetics*, Prentice Hall.

[46] Langsam, Y., Augenstein, M. J. and Tenenbaum, A. (1998) *Data Structures Using C*, Prentice Hall.

[47] Levinson, W. and Jawest, E. (2002) *Medical Biology & Immunology*, Examination & Board Review.

[48] Liang, S., Fuhrman, S. and Brown P.O (1998) *REVEAL, a general reverse engineering algorithm for inference of genetic network architectures*, Pacific Symposium on Biocomputing, 98, pp:18-29.

[49] Mestl, T., Lemay, C. and Glass, L. (1996) *Chaos in high-dimensional neural and gene networks*, Physica D 98, pp:33-52.

[50] Meyers, S., and Friedland, P. (1984) *Knowledge-based simulation of genetic regulation in bacteriophage lambda*, Nucleic Acids Res. 12, 1, pp:1-9.

[51] Miles, B. (2003) *Regulation of glycogen metabolism*, Lecture Notes, Texas A&M University.

[52] Monod J. (1965) *From enzymatic adaptation to allosteric transitions*, Nobel Lecture.

[53] Nguyan, D. V., Arpat, A. B., Wang, N. and Carroll, R. J. (2002) *DNA Microarray Experiments: Biological and Technological Aspects*, Biometrics 58, pp:701-717.

[54] Öktem, H., Pearson, R. and Egiazarian, K. (2003) *An adjustable aperiodic model class of genomic interactions using continuous time Boolean networks (Boolean delay equations)*, Chaos 13, 4, pp:1167-1174.

[55] Oppenheim, A.V. (1983) *Signals and Systems*, Prentice-Hall.

[56] Palis, J. and de Melo, W. (1982) *Geometric Theory of Dynamical Systems: An Introduction*, Springer-Verlag, NewYork.

[57] Pavlidis, P., Lewis, D.P. and Noble, W.S. (2002) *Exploring gene expression data with class scores*, Proceedings of the Pacific Symposium on Biocomputing, pp:474-485.

[58] Pearl, J. (1998) *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, San Francisco, CA.

[59] Pickl, S. W. (1999) *Der τ-value als Kontrollparameter - Modellierung und Analyse eines Joint - Implementation Programmes mithilfe der kooperativen dynamischen Spieltheorie und der diskreten Optimierung*, Shaker-Verlag, Aachen.

[60] Ramalingam G. and Reps, T. (1996) *On the computational complexity of dynamic graph problems*, Theoretical Computer Science, 158, pp:223-227.

[61] Sakamoto, E. and Iba, H. (2001) *Inferring a system of differential equations for a gene regulatory network by using genetic programming*, Proc. Congress on Evolutionary Computation, pp:720-726.

[62] Salgado, H., Zavaleta, A. S., Castro, S. G., Zarate, D. M., Peredo, E. D., Solano, F. S., Rueda, E. P., Martinez, C. B. and Vides, J.C. (2001) *RegulanDB*

(version 3.2): transcriptional regulation and operon organization in Escherichia coli K-12, Nucleic Acids, 29,1, pp:72-74.

[63] Sekimizu, T., Park, H. and Tsujii, J. (1998) *Identifying the interaction between genes and gene products based on frequently seen verbs in Medline abstracts*, Genome Informatics, 9, pp: 62-71.

[64] Smolen, P., Baxter, D. A. and Byrne, J. H. (2000) *Modeling transcriptional control in gene networks- methods, recent results, and future directions*, Bulletin of Molecular Biology 62, pp:247-292.

[65] Somogyi, R. and Sniegoski, C. (1996) *Modeling the complexity of genetic networks: understanding multigenic and pleiotropic regulation*, Complexity 1, pp:45-63.

[66] Spellman, P., Sherlock, G., Zhang, M., Iyer, V., Anders, K., Eisen, M., Brown, P., Botstein, D. and Futcher, B. (1998) *Comprehensive identication of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization*, Mol. Biol. Cell 9, pp:3273-3297.

[67] Standish, T. A. (1995) *Data Structures, Algorithms & Software Principles in C*, Addison Wesley.

[68] Stein, O. (2003) *On Bi-level Strategies in Semi-Infinite Programming*, Nonconvex Optimization and Its Applications, Kluwer Academic Publishers.

[69] Stephens, M., Palakal, M., Mukhopadhyay, S. and Raje, R. (2001) *Detecting gene relations from Medline abstracts*, Pacific Symposium on Biocomputing 6, pp:483-496.

[70] Thomas, R. (1978) *Logical analysis of systems comprising feedback loops*, J. Theor. Biol. 73, pp:631-656.

[71] Thomas, R. and Kaufman, M. (2001) *Multistationary, the basis of cell differentiation and memory, I. Structural conditions of multistationary and other nontrivial behavior*, Chaos 11, 1.

[72] Thomas, R. and Kaufman, M. (2001) *Multistationary, the basis of cell differentiation and memory, II. Logical analysis of regulatory networks in terms of feedback circuits*, Chaos 11, pp:170-179.

[73] Thomas, R. (2001) *Regulatory networks seen as asynchronous automata: A logical description*, Journal of Theor. Biol., 153, pp:1-23.

[74] Turchin, V. (1977) *The Phenomenon of Science, A Cybernetic Approach to Human Evolution*, Columbia University Press, New York.

[75] Turing, A. M. (1952) *The chemical basis of morphogenesis*, Proceedings of the Royal Philosophical Society of London, Series B: Biological Sciences 237, pp:37-72.

[76] Tyson, J. J., and Othmer, H.G. (1978) *The dynamics of feedback control circuits in biochemical pathways*, Prog. Theor. Biol. 5, pp:1-62.

[77] Vangpanitlerd, S. (1996) *On the stability of linear discrete time systems*, IEEE Transactions on Automatic Control, pp:207-208.

[78] Vidyasagur, M. (1970) *An extension of bounded-input-bounded-output stability*, IEEE Transactions on Automatic Control, pp:702-703.

[79] Voet, D. and Voet, J. (2003) *Fundamentals of Biochemistry*, Wiley, John & Sons, Incorporated.

[80] Watson, J. D. and Crick, F. H. C. (1953) *Molecular structure of nucleic acids: A structure of deoxyribose nucleic acid*, Nature, 4356, pp:731-737.

[81] Weaver, D. C., Workman, C. T. and Stormo, G. D. (1999) *Modeling regulatory networks with weight matrices*, Proceedings of the Pacific Symposium on Biocomputing 99, pp:112-123.

[82] Weber, G.-W. (2003) *Generalized Semi-Infinite Optimization and Related Topics*, Research and Exposition in Mathematics 29, Heldermann Verlag.

[83] Weiss, M. A. (1997) *Data Structures and Algorithm Analysis in C*, Addison Wesley.

[84] Wessels, L. F. A., Van Someren, E. P. and Reinders, M. J. T. (2001) *A comparison of genetic network models*, Pacific Symposium on Biocomputing 01, pp:112-123.

[85] West, D.B. (2001) *Introduction to Graph Theory*, Second Edition, Prentice Hall.

[86] Wong, L. (2001) *PIES, a Protein Interaction Extraction System*, Pacific Symposium on Biocomputing 01, pp:520-531.

[87] Yilmaz, F. B., Öktem, H. and Weber, G.-W. (2004) *Mathematical modeling and approximation of gene expression patterns and gene networks*, Preprint 22, Institute of Applied Mathematics, METU, submitted for publication to *Operations Research Proceedings 2004* (selected papers of Operations Research Conference, Tilburg, Netherlands).