INSTITUTO POTOSINO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA, A.C.

POSGRADO EN CIENCIAS APLICADAS

Modeling and Simulation of Genetic Regulation Systems

Tesis que presenta

Luis Adolfo Torres González

Para obtener el grado de

Doctor en Ciencias Aplicadas

En la opción de

Control y Sistemas Dinámicos

Codirectores de Tesis:

Dr. Haret Codratian Rosu Barbus

Dr. Julio Collado Vides

San Luis Potosí, S.L.P., Agosto 2007.

IPICYT

Constancia de aprobación de la tesis

La tesis "**Modeling and Simulation of Genetic Regulation Systems**" presentada para obtener el Grado de de Doctor en Ciencias Aplicadas en la opción de Control y Sistemas Dinámicos fue elaborada por **Luis Adolfo Torres González** y aprobada el **8 de Agosto de 2007** por los suscritos, designados por el Colegio de Profesores de la División de Matemáticas Aplicadas del Instituto Potosino de Investigación Científica y Tecnológica, A.C.

Dr. Haret Codratian Rosu Barbus
(Co-director de la tesis)

Dr. Gerardo Rafael Argüello Astorga
(Sinodal)

Dr. Gerardo Escobar Valderrama
(Presidente)

Dr. José Luis González Solís
(Secretario)

Créditos Institucionales

Esta tesis fue elaborada en las instalaciones de la División de Matemáticas Aplicadas del Instituto Potosino de Investigación Científica y Tecnológica, A.C., bajo la codirección del Dr. Haret Rosu Barbus y el Dr. Julio Collado Vides del Centro de Ciencias Genómicas de la UNAM-Cuernavaca

Durante la realización del trabajo el autor recibió una beca académica del Consejo Nacional de Ciencia y Tecnología (83523) y del Instituto Potosino de Investigación Científica y Tecnológica, A.C.

Instituto Potosino de Investigación Científica y Tecnológica, A.C.

Acta de Examen de Grado

El Secretario Académico del Instituto Potosino de Investigación Científica y Tecnológica, A.C., certifica que en el Acta 004 del Libro Primero de Actas de Exámenes de Grado del Programa de Doctorado en Ciencias Aplicadas en la opción de Control y Sistemas Dinámicos está asentado lo siguiente:

En la ciudad de San Luis Potosí a los 8 días del mes de agosto del año 2007, se reunió a las 17:00 horas en las instalaciones del Instituto Potosino de Investigación Científica y Tecnológica, A.C., el Jurado integrado por:

Dr. José Luis González Solís	**Presidente**	**U. de G.**
Dr. Gerardo Escobar Valderrama	**Secretario**	**IPICYT**
Dr. Gerardo Rafael Argüello Astorga	**Sinodal**	**IPICYT**
Dr. Haret-Codratian Rosu Barbus	**Sinodal**	**IPICYT**

a fin de efectuar el examen, que para obtener el Grado de:

DOCTOR EN CIENCIAS APLICADAS
EN LA OPCIÓN DE CONTROL Y SISTEMAS DINÁMICOS

sustentó el C.

Luis Adolfo Torres González

sobre la Tesis intitulada:

Modeling and Simulation of Genetic Regulation Systems

que se desarrolló bajo la dirección de

Dr. Haret-Codratian Rosu Barbus
Dr. Julio Collado Vides (UNAM)

El Jurado, después de deliberar, determinó

APROBARLO

Dándose por terminado el acto a las 18:35 horas, procediendo a la firma del Acta los integrantes del Jurado. Dando fé el Secretario Académico del Instituto.

A petición del interesado y para los fines que al mismo convengan, se extiende el presente documento en la ciudad de San Luis Potosí, S.L.P., México, a los 8 días del mes agosto de 2007.

Dr. Marcial Bonilla Marín
Secretario Académico

L.C.C. Ivonne Lizette Cuevas Velez
Jefa del Departamento de Asuntos Escolares

INSTITUTO POTOSINO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA, A.C.
IPICYT
SECRETARIA ACADEMICA

Dedicatorias

Agradezco a los Doctores Haret Rosu Barbus y Julio Collado Vides por su valiosa dirección durante mi formación doctoral.

A mis amigos y compañeros que siempre estuvieron aquí y en el momento preciso.

Abstract

As the Biology of information processing in the living cell shifts from study of single signal transduction pathways to increasingly complex regulatory networks, physical and mathematical models become indispensable tools. Detailed predictive models of large genetic networks could revolutionize the ways the researchers study complex diseases as well as the design of modern medication. In this thesis, an attempt is made to present two different levels of description in models of genetic networks. First we show the advantages of Monte Carlo simulations and second, we apply simple dynamical analysis to the field of elementary genetic circuits. In these sense, we work on two main problems at two different scales: i) the gene clustering process of large genetic network using Potts model and ii) the much better controlled dynamical behavior of small genetic circuits. With respect to the first issue, we analyze the clustering that can occur in such networks as revealed by the microarray gene expression data defining the edge lengths of the network. As a novel idea, we use entropy to impose a basic condition that all clustering algorithm must satisfy. Regarding our usage of the Potts spin model to the gene expression data points, we introduce a distance-depending interaction between neighboring points following ideas of Domany and his collaborators. Concerning the small genetic circuits, the potential employment of two Observers for monitoring these fundamental processes is developed in detail. It is worth mentioning that the published version of the latter development was included in the selected list of contemporary topical areas in biological physics studies of the Virtual Journal of Biological Physics research in August 2005 (Vol. 10, No. 3).

Keywords: Clustering, Gene regulation, Gene networks, Dynamical analysis

Resumen

Los modelos físicos y matemáticos son herramientas indispensables en el procesamiento de información biológica de la célula viva; desde una vía de señalización simple hasta una red compleja regulatoria. La predicción de modelos detallados de grandes redes podrá revolucionar la forma en que los investigadores estudian las enfermedades complejas; así como el diseño de nuevos medicamentos. En esta tesis se hace un esfuerzo para presentar dos diferentes escalas de redes de genes. Primero, mostramos las ventajas de la simulación Monte Carlo, y segundo, aplicamos un análisis simple dinámico en el campo de los circuitos genéticos elementales. En este sentido, trabajamos dos problemas principales en dos escalas diferentes de redes de genes: i) el proceso de clustering en redes genéticas usando el modelo de Potts y ii) el comportamiento dinámico controlado de pequeños circuitos de genes. Con respecto al primer inciso, analizamos el clustering de tales redes considerando los datos de expresión de microarreglos que definen las dimensiones en la red. Como una idea novedosa, introducimos la definición de entropía como una condición fundamental que todo algoritmo de clustering debe satisfacer. Considerando nuestro uso del modelo espinorial de Potts a la expresión de genes, introducimos una interacción dependiente de la distancia entre puntos vecinos siguiendo las ideas de Domany y colaboradores. Concerniente a las pequeñas redes de genes, se emplearon dos Observadores para monitorear el desarrollo de esos procesos fundamentales en detalle. Es importante mencionar que una de las publicaciones logradas por este trabajo fue incluida en la lista seleccionada de tópicos contemporáneos en áreas de física biológica de la revista de investigación Virtual Journal of Biological Physics en Agosto 2005 (Vol.10, Issue 3).

Palabras Clave: Clustering, Regulación de genes, Redes de Genes, Análisis Dinámico.

Contents

Chapter 1
Preface and General Introduction

The analysis of genetic regulatory networks will much benefit from the recent upscaling to the genomic level of the many experimental methods in molecular biology. In addition to high-throughput experimental methods, mathematical and bioinformatics approaches are indispensable for the analysis of genetic regulatory networks. Given the size and complexity of most networks of biological interest, an intuitive comprehension of their behavior is often difficult, if not impossible to obtain. In this thesis, the two principal approaches that have been mostly used will be reviewed and employed in some detailed examples: the gene regulation process of large genetic networks by means of the Potts model and the dynamical behavior of small genetic circuits that are not so complicated and allow the application of control theory methods.

It is now commonly accepted that most interesting properties of an organism emerge from interactions between its genes, proteins, metabolites, and other secondary constituents. This implies that, in order to understand the functioning of an organism, we need to elucidate the networks of interactions involved in gene regulation, metabolism, signal transduction, and other cellular and intercellular processes.

Genetic regulatory networks control the spatiotemporal patterns of expression of genes in an organism, and thus underlie complex processes like cell differentiation and development of the tissues in prokaryotic and eukaryotic organisms. Genetic regulatory networks consist of genes, proteins, metabolites, and other small molecules, in the real of their mutual interactions. Their study has taken a qualitative leap through the usage of modern genomic techniques that allow simultaneous measurement of the expression levels of all genes of a given organism. In addition to experimental tools, mathematical methods supported by computer means are indispensable for the analysis of genetic regulatory networks since modeling and simulation procedures allow the behavior of large and complex systems to be predicted in a systematic way.

The order of the resulting papers in this thesis is the following: Chapter 4 – paper sent to reviewing; Chapter 5 – L.A. Torres, V. Ibarra-Junquera, P.Escalante-Minakata and H.C. Rosu., Physica A 380, 235-240 (2007); Chapter 6 – V. Ibarra Junquera, L.A. Torres, H.C. Rosu, G. Argüello and J. Collado Vides. Physical Review E 72, 011919 [10 pages] (2005).

Chapter 2
The Biology of Gene Expression

The future success of System Biology requires the establishment of general principles that can be used to link the behavior of individual molecules to system characteristics and functions. In order to achieve this goal and to study the different levels of description of genetic networks is necessary to show the basic principles of biology gene expression. The sophistication of biological control systems is extraordinarily rich and regulation takes place on many different levels simultaneously. Novel surprising details are constantly revealed and new technologies are invented (microarray). This chapter will focus on regulatory processes at the level of gene transcription. We will briefly summarize some basic concepts from molecular level biology and then discuss some of the general principles involved in the regulation of gene transcription. This discussion will be augmented by a walk-through of some of the best studied natural gene regulatory systems. The purpose of this section is to provide a brief introduction to the fundamental biology of gene expression.

2.1 The Genetic Code

Most regulatory processes that take place within cells involve proteins whose structure and function is determined by information stored in the cell's DNA. Genetic engineered and the engineering of gene networks involve the manipulation of this information and of the conditions under which it is used to synthesize proteins. The DNA molecule encodes information in the four nucleotides containing the bases adenine (A), guanine (G), cytosine (C) and thymine (T). The molecular structure of the nucleotides is illustrated in Fig. 1.2A The carbon atoms are indicated as solid circles, lines indicate covalent bonds between atoms and sticks indicate a covalent bond that ends in hydrogen. RNA and DNA differ in the identity of the atom bound to the carbon at position 2' in the sugar ring (marked by an X). RNA has a hydroxyl group bound at this position while DNA has a hydrogen atom. Polynucleotide chains are formed by individual ribonucleotides being linked o each other through a phosphodiester

bond [1]. This bond is between the phosphate group bound to the carbon at position 5' and the oxygen bound to the carbon at position 3' and establishes the 5' and establishes the 5' → 3' directionality of the polymer chain. Under normal conditions, the DNA is in a double stranded form that consists of the 5' - 3' strand and its complement where the direction of the DNA backbone is reversed (Fig 1.2B) Bases on opposite strands are paired with each other through hydrogen bonds such that A couples with T and C couples with G. The double stranded DNA forms a helical structure (Fig **1.2C**).

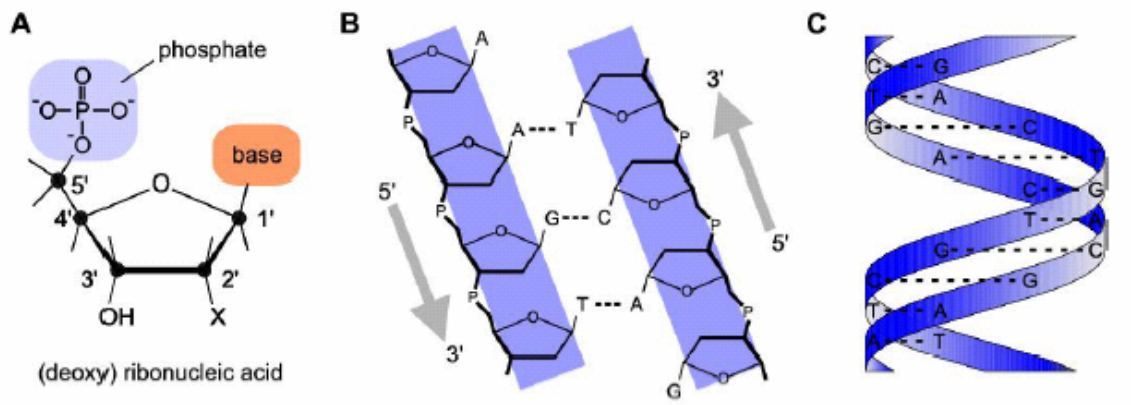


Figure 1.2 (A) The molecular structure of ribonucleic acid (RNA) and deoxyribonucleic acid (DNA). RNA and DNA has a hydroxyl group and a hydrogen atom at position X, respectively. In DNA, the base bound to the carbon at position 1' is adenine, guanine, cytosine or thymine. In RNA, the thymine is replaced by Uralic. **(B)** Double stranded DNA. Hydrogen bonds (broken line) are formed between the bases A and T or G and C and links to gather two complementary single stranded DNA molecules. **(C)** The helical structure of double stranded DNA.

Transcription

The synthesis of a protein based on the DNA-encoded amino acid sequence requires at least two steps. First, the genomic information must be transcribed from the DNA sequence into a messenger RNA molecule (mRNA). This is done by an RNA polymerase, which, in analogy to DNA polymerase, catalyzes the formation of phosphodiester bonds between individual nucleotides. The structure of RNA molecules is similar to that of DNA molecules with the exception that the backbone consists of ribose rather than deoxyribose and the base thymine is replaced by the base uracil (U), Furthermore, the mRNA is usually single stranded [1-2].

Translation

After transcription, the message contained in the mRNA must be translated into a protein. This is done by the ribose, which is a molecular machine made of both RNA and protein. The process of translation involves two additional types of RNA molecules, ribosomal RNA (rRNA) and transfer RNA (tRNA). The rRNA molecules are components of the ribosome. The tRNA provides the specificity that enables the insertion of the correct amino acid into the protein that is being synthesized.

Proteins

Proteins consist of a chain in which individual amino acids residues are linked to each other through peptide bonds. The general structure of the amino acids is illustrated in Fig. **2.2A**. In analogy with DNA and RNA, they consist of a common element that enables the formation of a polymer chain. The identity and the property of the individual amino acids are determined by the side chain. There are 20 naturally occurring amino acids. In the polymer chain that forms the backbone of proteins, the individual amino acids are linked to each other through peptide bonds formed between the carboxyl-group of the amino acid and the amino-group of another (Fig **2.2B**). This creates a chain that at one end has a free amino-group, the N-terminal (NH^+_3), and the other end has a free carboxyl-group, the C-terminal (COO^-).

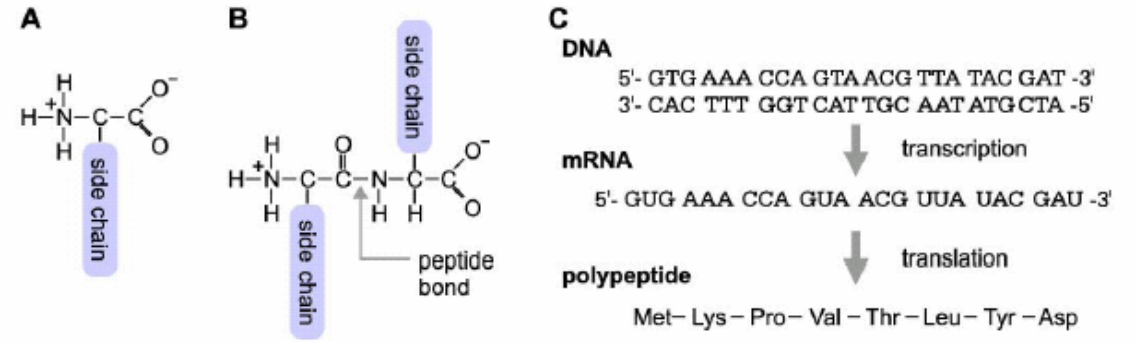


Figure 2.2 (A) The molecular structure of amino acids. The identity of amino acid is determined by its side chain. **(B)** Peptide bond formed between the amino- and the carboxyl-groups of two amino acids. **(C)** The correspondence between the DNA sequence, mRNA sequence and that sequence of the first eight amino acids of the LacR repressor protein.

Codons

The DNA molecule stores the information required to synthesize proteins in terms of a string of codons. A codon consists of three nucleotides, each selected from of the four available bases (A, T, G or C) which are read from the DNA molecule in the 5' to 3' direction. In Fig **2.2B** the codon encoded on the left is AGT while the codon encoded on the right strand is ACT. Of the 64 possible codons, 61 encoded for one o the 20 amino acids (Table **1.2**) The genetic code is thus redundant and different codons may identify the same amino acid. The last three codons (TAA, TAG and TGA) are stop codons. They define the end of the protein encoding region of the DNA. In addition, the order of the amino acids in the polypeptide chain is determined by the sequence in which the codons appear in the DNA sequence. In most cases, there is a linear relationship between the DNA sequence and the amino acid sequence within the protein that the sequence encodes. Fig **2.2C** shows the first 24 base pairs of the gene that encodes the LacR repressor protein, the corresponding mRNA sequence and the sequence of the first 8 amino acids in the LacR repressor polypeptide chain. The N- and C- terminal regions are encoded by the codons in the 5' and the 3' end of the DNA-encoding sequence, respectively.

1st (5')	2nd T	C	A	G	3rd (3')
A	Isoleusine	Threonine	Lysine	Arginine	A
A	Isoleusine	Threonine	Asparagine	Serine	T
A	Isoleusine	Threonine	Asparagine	Serine	C
A	Methionine	Threonine	Lysine	Arginine	G
T	Leucine	Serine	STOP	STOP	A
T	Phenylalanine	Serine	Tyrosine	Cysteine	T
T	Phenylalanine	Serine	Tyrosine	Cysteine	C
T	Leucine	Serine	STOP	Tryptophan	G
C	Leucine	Proline	Glutamine	Arginine	A
C	Leucine	Proline	Histidine	Arginine	T
C	Leucine	Proline	Histidine	Arginine	C
C	Leucine	Proline	Glutamine	Arginine	G
G	Valine	Alanine	Glutamic acid	Glycine	A
G	Valine	Alanine	Apartic acid	Glycine	T
G	Valine	Alanine	Apartic acid	Glycine	C
G	Valine	Alanine	Glutamic acid	Glycine	G

Table 1.2 The correlation between the sequence of bases in the codons and the amino acids. The codon TAA, TGA and TAA signals termination of translation.

Once translation is completed and the full length DNA-encoded polypeptide has been formed, the function of many proteins requires the completion of additional steps. This may involve, for example, covalent modification, such as phosphorylation, acetylation or glycosylation, i.e. the addition of a phosphate, an acetyl or a glycosyl-group, the incorporation of the protein to its appropriate cellular location, for instance, in the cell membrane [1].

2.2 Genes and Gene Expression

The term gene is usually used to refer to the *DNA sequence that is transcribed into mRNA and subsequently translated into a protein*. However, there are important exceptions to this rule. For example, DNA sequences that encode for molecules like rRNA and tRNA are genes even though the RNA molecule is never translated into a protein. Genes are usually carried on the cell's chromosomes. Each chromosome carries at least one origin of replication. These regions determine the location where the DNA polymerase initiates the duplication of the genetic material. The location of a specific gene on the chromosome is called the gene's locus. Haploid cells carry a single copy of each chromosome and the locus thus uniquely determines the location of the gene. Diploid cells have homologous chromosome pairs. Two different forms of the same gene are known as alleles [2-3].

The chromosomes are organized very differently in prokaryotic, which lack a cell nucleus, and in eukaryotic cells. In bacteria (a prokaryote), such as Escherichia coli, all of the genes are located on a single, circular chromosome while the genes in eukaryotic cells are located on several linear chromosomes. There are 16 chromosomes in yeast. In addition, the eukaryotic DNA is complexed with nuclear proteins and compacted into a structure called chromatin. Central to this structure is the wrapping of approximately 200 base pairs of DNA around protein complexes known as nucleosomes (Fig. **3.2A**). The organization of chromatin and of the nucleosomes can be used as an instrument to regulate which genes are accessible for transcription by RNA polymerase. The primary constituent of the nucleosomes is the four histone proteins H2A, H2B, H3 and H4, which combine to form a histone tetramer (Fig **3.2B**). A nucleosome consists of two histone tetramer. Each histone subunit has a protruding N-terminal "tail" that serves important regulatory functions. There, covalent modifications, such as acetylation, can greatly influence the accessibility of the DNA. The nucleosomes are, together with other nuclear proteins, arranged into chromatin fibers. Examples of potential spatial arrangements of the nucleosomes are shown fig **3.2C**. In addition to the chromosomes, genes can be carried on plasmids. Plasmids are in many ways similar to the bacterial chromosome. They are circular pieces of DNA that typically replicate independently of duplication of the chromosomal DNA prior to cell division. As a result, plasmids are often present in multiple copies within each cell and the plasmid copy number usually changes as cells progress through the cell division cycle. The average copy number of plasmid per cell depends on the type of the origin of replication that it carries. Some plasmids are stringently controlled and are present only in a single copy while others are loosely regulated and present in 60 copies per cell or higher [1-3].

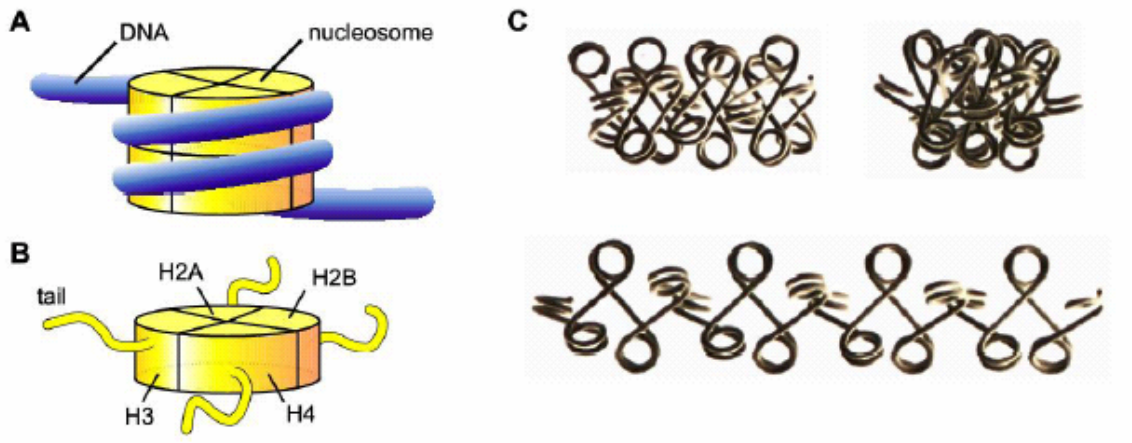


Figure 3.2 (A) Schematic illustration of DNA wrapped around a nucleosome. **(B**) The primary component of the nucleosomes consists of four histone proteins H2A, H2B, H3 and H4. The nucleosomes can be remodeled and rearranged spatially by covalent modification of the protruding histone tails. **(C)** Illustration of potential organizations of nucleosomes in spatial structures.

In addition to the sequences that encode for genes, the DNA contains regions that are involved in the regulation of gene transcription. The RNA polymerase reads the genetic code in the 5´ to 3´ direction and the location where it initially binds to the DNA is located upstream of the gene, i.e., farther in the 5´ direction (Fig. 4.2). The region where the RNA polymerase initially contacts the DNA is called the promoter of the gene whose expression it facilitates. The expression of a gene may occur from more that one promoter, i.e., the region upstream of the gene may contain distinct binding sites for the RNA polymerase. The first nucleotide that is transcribed is usually labeled +1 and nucleotides are counted relative to this transcription start site in the 5´ to 3´direction of the DNA. The nucleotides in the gene-encoding region are thus labeled with positive numbers while the promoter regions are labeled with negative numbers. In bacteria, the promoter region is about 60 base pairs in length and spans roughly 40 base pair upstream and roughly 20 base pairs downstream of the +1 site. In years, the promoter region spans roughly 200 base pairs.

Generally speaking, no two promoters are identical. Statistical analysis has however shown that there are regions that are highly conserved within different promoters. In bacteria, one of these regions is located at position -10 and has the consensus sequence TATAAT. This region is called the *TATA-box* and is in many cases essential for the proper alignment of the RNA polymerase holoenzyme with respect to the gene encoding sequence. Mutations of the TATA-box sequence, i.e., the substitution of one nucleotide with another, can greatly affect the rate at which the DNA is transcribed into an mRNA. A sequence that is similar to the TATA-box is also important for the transcription of many eukaryotic genes [3].

In addition to the TATA-box, the promoter region often contains sites where transcription factor proteins can bind and directly or indirectly affect the rate of transcription. In bacteria, transcription factor binding sites are often referred to as *operators*.

However, such regulatory elements may also be located far from the promoter region or even within the gene-encoding region of the DNA. In eukaryotes, it is quite common to find *enhancer sequences* that affect the transcription from a promoter located very far from it in the DNA sequence. This action-at-a-distance can arise from the rearrangement of chromatin structure and/or close spatial proximity of transcription factors bounds to the enhancer sequence due to bending and looping of the DNA. Transcription factor binding sites are referred to as *cis*-regulatory elements while the transcription factor proteins that bind to them are referred to as *trans*-regulatory elements.

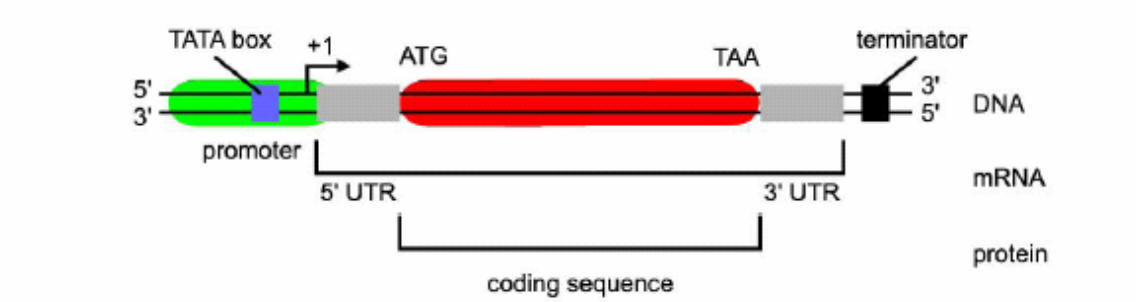


Figure 4.2 Typical organization of a gene containing the information required for the synthesis of a protein. The promoter is the region where the RNA polymerase initially binds. The terminator is the region where the RNA polymerase is released from the DNA. The DNA also contains regions that, when transcribed into mRNA, controls translation initiation (5' UTR) and termination (3' UTR).

In addition to the promoter and cis-regulatory elements, there are sequences within the DNA that determine the termination of transcription (a terminator sequence) and, for protein-encoding genes, sequences that determine the region of the mRNA that is to be translated into protein (Fig. 4.2). The codon that indicates the location where translation is to start, the translation start codon, is often ATG. The DNA sequence located between the start site of transcription and start of translation is referred as an untranslated region (UTR). UTRs can greatly influence the efficiency of gene expression, for example by determining how well the ribosomes can bind to the mRNA and initiate translation. The translation stop codon that indicates the location where translations is terminated is either TAA, TAG or TGA. The sequence of the DNA between the stop codon and the site where transcription is terminated can also have an effect on the efficiency of gene expression. This region is referred to as the 3´UTR [3].

2.3 Transcription and Translation

Similarly to the definition of a gene, the meaning of gene expression is not always clearly defined. Some use the term gene expression to refer to the biological manifestation in terms of alteration in phenotype, that is, an observable change in the characteristics of the cell. The gene that is responsible for a specific cellular that can be said to be expressed when the phenotype is observed and not expressed otherwise. In other words, gene expression can be viewed as being a binary on/off process. Others use gene expression to refer to the process that starts when the transcription of the DNA that encodes the gene is initiated and ends when a biologically functional molecule is formed, regardless of whether this is accompanied with a detectable change in the cell's phenotype. In this view, gene expression can be graded and quantified based on measurements of the activity of the end product of the gene expression process.

Since many proteins require some post-translational modification to be fully functional, e.g., the attachment of a phosphate group or the incorporation of the protein into a larger complex, it can be argued that such events are part of the process in which the genomic information is expressed. Generally speaking,

however, there will be a positive correlation between the rate at which a gene is transcribed and the abundance (and hence the activity) of the end product of the gene expression process. Typically, if a gene's mRNA is abundant within a cell, there will be a high level of the corresponding protein product. Transcription is usually a prerequisite for gene expression and the control of transcription is one of the most important regulatory instruments available to the cells. In prokaryotes as well as eukaryotes, the transcription of a gene into a corresponding mRNA occurs in three general steps: transcription initiation, elongation of the mRNA and termination of transcription. Gene expression can be regulated on all of these levels. Regulation of gene expression at the levels of transcription initiation is, however, the most common.

2.4 Regulation of Gene Expression

The means employed to regulate gene expression are remarkable and many. The most obvious method of control, and the one that can be most readily manipulated, is the modulation of the frequency of transcription initiation. The next sections will discuss how this method of gene expression control is utilized in three well-studied systems; the lactose operon in *E. coli*, the λ CI repressor in bacteriophage λ and the galactose utilization network in *Saccharomyces cerevisiae* [4].

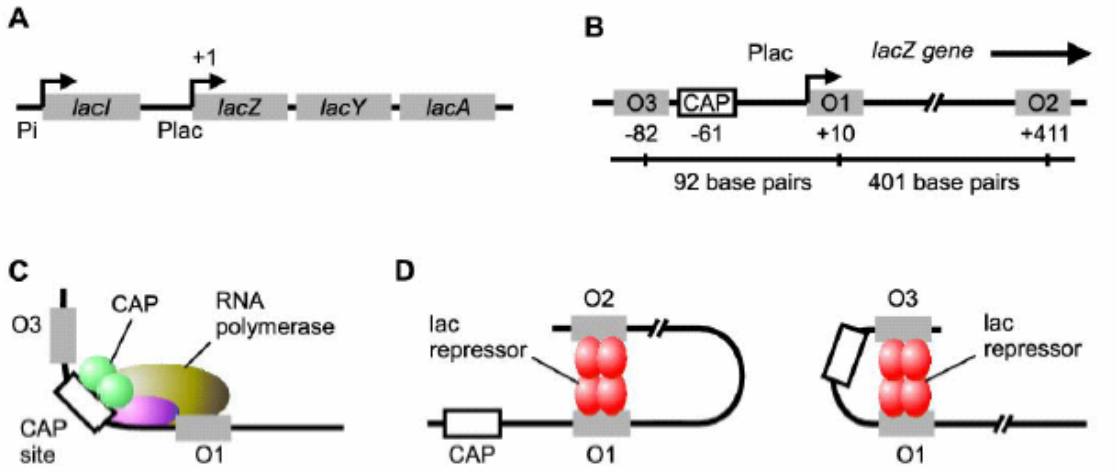


Figure 5.2 (A) The genes *lacZYA* of the lactose operon share the same promoter, P_{lac}, which is repressed by the repressor encoded by the adjacently located lacI gene. **(B)** Regulatory elements of the P_{lac} promoter. The LacR repressor can bind to the three *lacO* operators O1, O2 and O3. The CAP protein can bind to the CAP operator. **(C)** Activation of transcription by CAP. **(D)** Repression involves DNA looping facilitated by LacR repressor tetramers bound to different operator sites.

2.4.1 The Lactose Operon of *E. coli*

The lactose operon in *E. coli* consists of three genes, *lacZ*, *lacY* and *lacA*, whose transcription is initiated from a single promoter region, P_{lac} (Fig. **5.2A**). The rate of transcription of the *lacZYA* genes is regulated by the LacR repressor

protein and by a protein called CRP (Cyclic AMP receptor protein) or CAP (cAMP activating protein). CAP can act as a transcriptional activator. It binds as a dimmer to an operator site centered at position -61 relative to the transcription start site (Fig. **5.2B**). It affects the process of transcription initiation by interacting directly with the α-subunit of the RNA polymerase holoenzyme. It has been observed that the presence of CAP increases the amount of the open complex some 13-fold, but that its presence does not change the rate of the transition between the closed and the open complex. This indicates that CAP may act at the first step in transcription initiation by increasing the rate at which the holoenzyme binds to the promoter and/or by decreasing the rate at which the holoenzyme dissociates from the promoter [4-5].

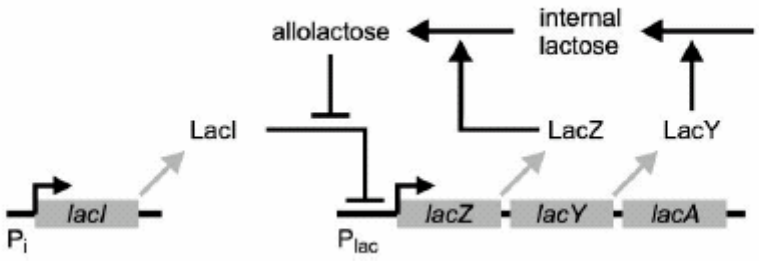


Figure 6.2 Feedback regulation of the lactose operon. Allolactose inhibits the activity of the repressor and relieves its effect on the transcription of the lactose operon genes. This causes upregulation of lacZ and lacY, which, in turn, causes an increased rate of allolactose production and lactose uptake, respectively.

The LacR repressor protein is, as the name implies, an inhibitor of transcription of the genes in the lactose operon. It is expressed *constitutively*, i.e., at a constant rate, from the P_i promoter and is located adjacent to the lactose operon (Fig. **5.2A**). The LacR protein binds as a tetramer to three *lacO* operators, O1, O2 and O3, centered at positions +11, -82 and +410, respectively (Fig. **5.2B**). The operators have nearly palindromic sequences and are composed of two half-sites that each makes contact with one LacR monomer in the tetrametric repressor complex. It is believed that the binding of the LacR repressor to O1 prevents the binding of the RNA polymerase holoenzyme to the promoter through *steric hindrance*; the repressor tetramer may simply act as a space-excluding barrier for the incoming holoenzyme. Elimination of the auxiliary *lacO* operators O2 and O3 does not abolish the inhibitory function of LacR, but reduces its effect. While elimination of either O2 or O3 causes a 3-fold reduction in repression, eliminating both causes a 70-fold reduction. Thus, the auxiliary operators appear to serve redundant roles in the inhibition of transcription by the LacR protein. The efficient repression observed in the presence of two or three of the operators are believed to be due to looping of the DNA. The binding of the repressor tetramer to a single operator involves only two of its four subunits, which leaves two subunits capable of binding a second operator site provided that the DNA is twisted into a loop structure (Fig. **5.2D**). These loop structures may act as barriers that limit the accessibility to the promoter region and/or as a roadblock of its movement along the DNA.

The above discussion of the regulation of the lactose operon addresses the interaction between *cis*- and *trans*-regulatory elements in the promoter region. In addition to this, the activity of the *trans*-factors, i.e., CAP and the LacR repressor, are extensively regulated. First of all, the activity of CAP depends on the presence of cAMP. The concentration of cAMP in turn depends on the presence of glucose. The transcription of the genes in the lactose operon is negatively correlated with the concentration of glucose in the growth medium. CAP affects the transcription of a large number of genes and is a central player in the global gene regulatory system known as catabolite repression. This system ensures that the cell does not wastefully express the genes required for metabolizing other sugars when the energy-rich glucose in available [3-6].

The activity of the lactose operon is modulated via a feedback loop involving the proteins LacR, LacZ and LacY (Fig. **6.2**). The genes *lacZ* and *lacY* encodes for the enzyme β-galactosidase and the membrane-bound lactose permease, respectively. While the lactose permease enables the transport of extracellular lactose into the cell, the β-galactosidase converts intracellular lactose into glucose and galactose. It also converts some of the lactose into allolactose. Allolactose in turn binds to the LacR tetramer and causes a conformational change, or *allosteric transition*, to a state that has a significantly reduced affinity for the operator sites. As a result, the presence of small amounts of the allolactose, the *inducer* of LacR, causes an up-regulation of the expression of the *lacZYA* genes in the lactose operon. This causes an increased rate of lactose uptake (by LacY) and conversion of lactose into allolactose (by LacZ), which, in turn, lowers the activity of LacR even further. The lactose operon is thus regulated through a positive feedback loop and catabolite repression. This enables an energy-efficient switch. The *lacZYA* genes are expressed at low (basal) levels when glucose is present and are only activated when needed, i.e., when glucose is absent and lactose is present. Many other operons are regulated in a manner that resembles that of the lactose operon and it is a textbook example of a simple gene regulatory circuit.

2.5 DNA Microarrays

The DNA microarray technology has received a great deal of attention in the last few years. Advanced computational methods are constantly improving, aiming to analyze and interpret the enormous amount of gene expression data. The DNA-chip method is a powerful, flexible and relatively simple procedure. Unlike traditional methods in molecular biology, which generally work on one or few genes per experiment, the DNA-chip method enables the monitoring of the expression level of hundreds to thousands of genes in a parallel way [7].
Variation in DNA sequence underlies most of the differences we observe within and between species. Locating, identifying, and cataloguing these genotypic differences represent the first steps in investigating the genomic variation among and within living organisms.
Changes in multigene patterns of expression can provide clues about cellular functions and biochemical pathways, as well as discovery of new, interesting genes, which may be potential markers for diagnosis or playing a role in drug

therapy. The improvement in DNA-chip technology, together with increasing genome-sequence information for different organisms, including humans, will enable even higher levels of quality and complexity of microarray experiments.

The principle of a microarray experiment, is that mRNA from a given cell line or tissue is used to generate a target which is hybridized in parallel to a large number of DNA sequences, immobilized on a solid surface in an ordered array [8].

The most commonly used procedures of this type can be divided into two groups, according to the arrayed material: complementary DNA (cDNA) and oligonucleotide microarrays. The first group allows comparison of fluorescently labeled cDNA populations from control and experimental tissues, marked by two colors. This technique is flexible in the choice of arrayed elements, particularly in preparation of small, customized arrays for specific investigation.

Microarray experiments are used to quantify and compare gene expression on large scale. cDNA microarrays consist of thousands of individual DNA sequences printed in a high-density array on a glass microscope slide by a robotic arrayer. The relative abundance of the spotted DNA sequences in two samples can be assessed by monitoring the differential hybridization of the two samples, to the sequences on the array. For mRNA samples , the two samples or targets are reverse transcribed into cDNA, labeled using different fluorescents dyes (usually a red-fluorescent dye, Cyanine 5 (Cy5), and a green-fluorescent dye, Cyanine 3 (Cy3)), then mixed in equal proportions and hybridize with the arrayed DNA sequences or probes. After this competitive hybridization, the slides are imaged using a scanner, and fluorescence measurements are made separately for each dye at each spot of the array. The ratio of the red and green fluorescence intensities for each spot is indicative of the relative abundance of the corresponding DNA probe in the two nucleic acid target samples [9].

References

[1] Alberts B. *et al*., The Molecular Biology of the Cell, Garland Science. New York, New York (2002).

[2] Latchman D. Gene Regulation: A Eukaryotic Perspective. Stanley Thornes. Cheltenham. United Kingdom (1998).

[3] Lewin B. Genes VII. Oxford University Press. Oxford, United Kingdom (2000).

[4] Müller-Hill B. The lac Operon. De Gruyter, Germany (1996).

[5] Ptashne M., & Gann A.. Genes & Signals. Cold Spring Harbor Laboratory Press. Cold Spring Harbor, New York (2002).

[6] White R.J. Gene Transcription: Mechanism and Control. Blackwell Science. Oxford, United Kingdom (2001).

[7] Gerhold, D., T. Rushmore, and C.T. Caskey*,* DNA chips: promising toys have become powerful tools*. Trends Biochem Sci,* **24**(5), 168-73 (1999).

[8] Schulze, A. and J. Downward, Navigating gene expression using microarrays a technology review. *Nat Cell Biol*. **3**(8), 190-5 (2001).

[9] Hwa Yang Y. and Speed Terry. Design issues for cDNA Microarray Experiments. *Nature Reviews Genetics*. **3**, 579-588 (2002).

Chapter 3
Modeling and Simulation of Genetic Regulatory Networks

In order to understand the functioning of organisms on the molecular level, we need to know which genes are expressed, when and where in the organism, and to which extent. The regulation of gene expression is achieved through genetic regulatory systems structured by networks of interactions between DNA, RNA, proteins, and small molecules. As most genetic regulatory networks of interest involve many components connected through interlocking positive and negative feedback loops, loops, an intuitive understanding of their dynamics is hard to obtain. As a consequence, formal methods and computer tools for the modelling and simulation of genetic regulatory networks will be indispensable. This chapter reviews some formalism that have been employed in mathematical biology and bioinformatics to describe genetic regulatory systems, in particular, clustering and Potts model. In addition, this chapter shows how these formalisms have been used in the simulation of the behaviour of actual regulatory systems.

3.1The role of computation and mathematics in complex networks

The genome of an organism plays a central role in the control of cellular processes, such as the response of a cell to environmental signals, the differentiation of cells and groups of cells in the unfolding of developmental programs, and the replication of the DNA preceding cell division. Proteins synthesized from genes may function as transcription factors binding to regulatory sites of other genes, as enzymes catalyzing metabolic reactions, or as components of signal transduction pathways. With few exceptions, all cells in an organism contain the same genetic material. This implies that, in order to understand how genes are implicated in the control of intracellular and intercellular processes, the scope should be broadened from sequences of nucleotides coding for proteins to regulatory systems determining which genes are expressed, when and where in the organism, and to which extent.

Gene expression is a complex process regulated at several stages in the synthesis of proteins. Apart from the regulation of DNA transcription, the best

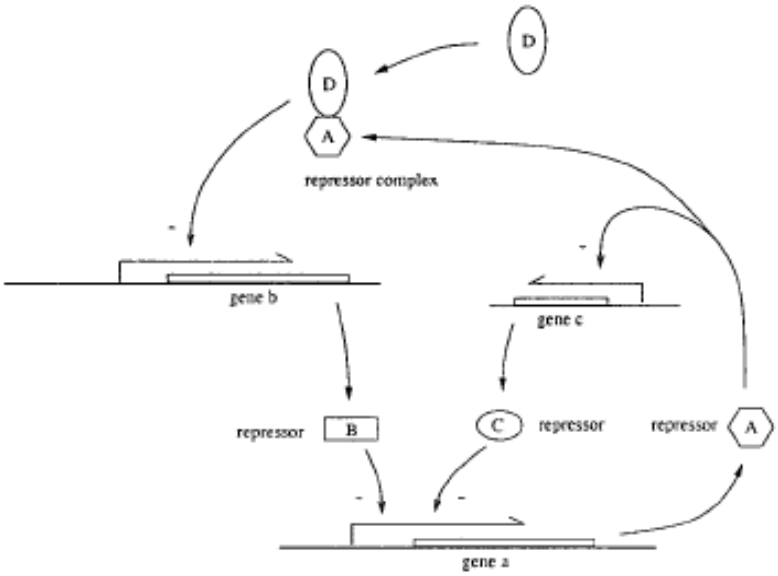


Figure 1.3 Example of a genetic regulatory system, consisting of a network of three genes a, b, and c, repressor proteins A, B, C, and D, and their mutual interactions. The figure distinguishes several types of interactions.

studied form of regulation, the expression of a gene may be controlled during RNA processing and transport (in eukaryotes), RNA translation, and the posttranslational modification of proteins. The degradation of proteins and intermediate RNA products can also be regulated in the cell. The proteins fulfilling the above regulatory functions are produced by other genes. This gives rise to genetic regulatory systems structured by networks of regulatory interactions between DNA, RNA, proteins, and small molecules. An example of a simple regulatory network, involving three genes that code for proteins inhibiting the expression of other genes, is shown in Fig. **1.3**. Proteins B and C independently repress gene *a* by binding to different regulatory sites of the gene, while A and D interact to form a heterodimer that binds to a regulatory site of gene *b* [1]. Binding of the repressor proteins prevents RNA polymerase from transcribing the genes downstream. Analyses of the huge amounts of data made available by sequencing projects have contributed to the discovery of a large number of genes and their regulatory sites. The KEGG database, for instance, contains information on the structure and function of about 110,000 genes for 29 species [1-2]. In some cases, the proteins involved in the control of the expression of these genes, as well as the molecular mechanisms through which regulation is achieved, have been identified. Much less is known, however, about the functioning of the regulatory systems of which the individual genes and interactions form a part [3-13]. Gaining an understanding of the emergence of complex patterns of behavior from the interactions between genes in a regulatory network poses a huge scientific challenge with potentially high industrial pay-offs. The study of genetic regulatory systems has received a major impetus from the recent development of experimental techniques like cDNA microarrays and oligonucleotide chips, which permit the spatiotemporal expression levels of genes to be rapidly measured in a massively parallel way [14-17]. Other techniques, such as the mass spectrometric identification of gel-separated proteins, allow the state of a cell to be characterized on the proteomic level as well [18-21]. Although still in their infancy, these techniques

have become prominent experimental tools, by opening up a window on the dynamics of gene expression. In addition to experimental tools, formal methods for the modeling and simulation of gene regulation processes are indispensable. As most genetic regulatory systems of interest involve many genes connected

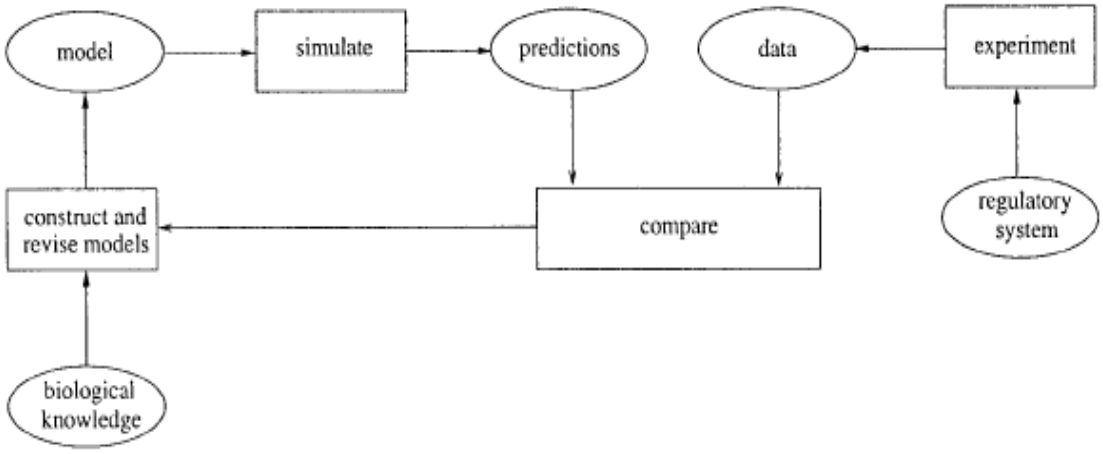


Figure 2.3 Analysis of genetic regulatory systems. The boxes represent activities, the ovals information sources.

through interlocking positive and negative feedback loops, an intuitive understanding of their dynamics is hard to obtain. Using formal methods, the structure of regulatory systems can be described unambiguously, while predictions of their behavior can be made in a systematic way. Especially when supported by userfriendly computer tools, modeling and simulation methods permit large and complex genetic regulatory systems to be analyzed. Figure **2.3** shows the combined application of experimental and computational tools. Starting from an initial model, suggested by knowledge of regulatory mechanisms and available expression data, the behavior of the system can be simulated for a variety of experimental conditions. Comparing the predictions with the observed gene expression profiles gives an indication of the adequacy of the model. If the predicted and observed behavior does not match, and the experimental data is considered reliable, the model must be revised. The activities of constructing and revising models of the regulatory network, simulating the behavior of the system, and testing the resulting predictions are repeated until an adequate model is obtained.

The formal basis for computer tools supporting the modeling and simulation tasks in Fig. **2.3** lies in methods developed in mathematical biology and bioinformatics. Since the 1960s, with some notable precursors in the two preceding decades, a variety of mathematical formalisms for describing regulatory networks have been proposed. These formalisms are complemented by simulation techniques to make behavioural predictions from a model of the system, as well as modeling techniques to construct the model from experimental data and knowledge on regulatory mechanisms. Traditionally, the emphasis has been on simulation techniques, where the models are assumed to have been hand-crafted from the experimental literature. With more experimental data becoming available and easily accessible through databases

and knowledge bases, modeling techniques are currently gaining popularity. This chapter gives an overview of two formalisms to describe genetic regulatory networks and discusses their use in the modeling and simulation of regulatory systems. Recently, a collection of introductory chapters covering some of the methods that include directed graphs, Bayesian networks, Boolean networks and their generalizations, ordinary and partial differential equations, qualitative differential equations, stochastic master equations, and rule-based formalisms [22-30] . Moreover, in the last few years the number of papers seems to be growing in an exponential fashion.

3.2 The different levels of description in models of genetic networks

As the biology of information processing in the living cell shifts from the study of single signal transduction pathways to increasingly complex regulatory networks, mathematical models become indispensable tools. Detailed predictive models of large genetic networks could revolutionize how researchers study complex diseases, yet such models are not yet within reach. One reason is that experimental data for large genetic systems are incomplete; another is that large genetic systems are difficult to model [31]. Extrapolating the standard differential equations model of a single gene (with its several kinetic parameters) to large systems would render the model prohibitively complicated. One possible way to simplify such models would be to find a “coarse grained” level of description for genetic networks; that is, to focus on the system behavior of the network while neglecting molecular details wherever possible (Fig. **3.3**). Such an approach exists for other fields of science, for example, the concept of molecular orbitals in organic chemistry, which mercifully spares us from the details of the underlying quantum physics. Brandman *et al*. points to the possibility of simplifying large genetic networks models [32]. Using a standard differential equations approach, the authors find that the intricate internal dynamics of a frequent cellular subcircuit exhibits a simple bistable “ON/OFF” behavior, and thus could be modeled by something much simpler than differential equations, something as simple as a switch.

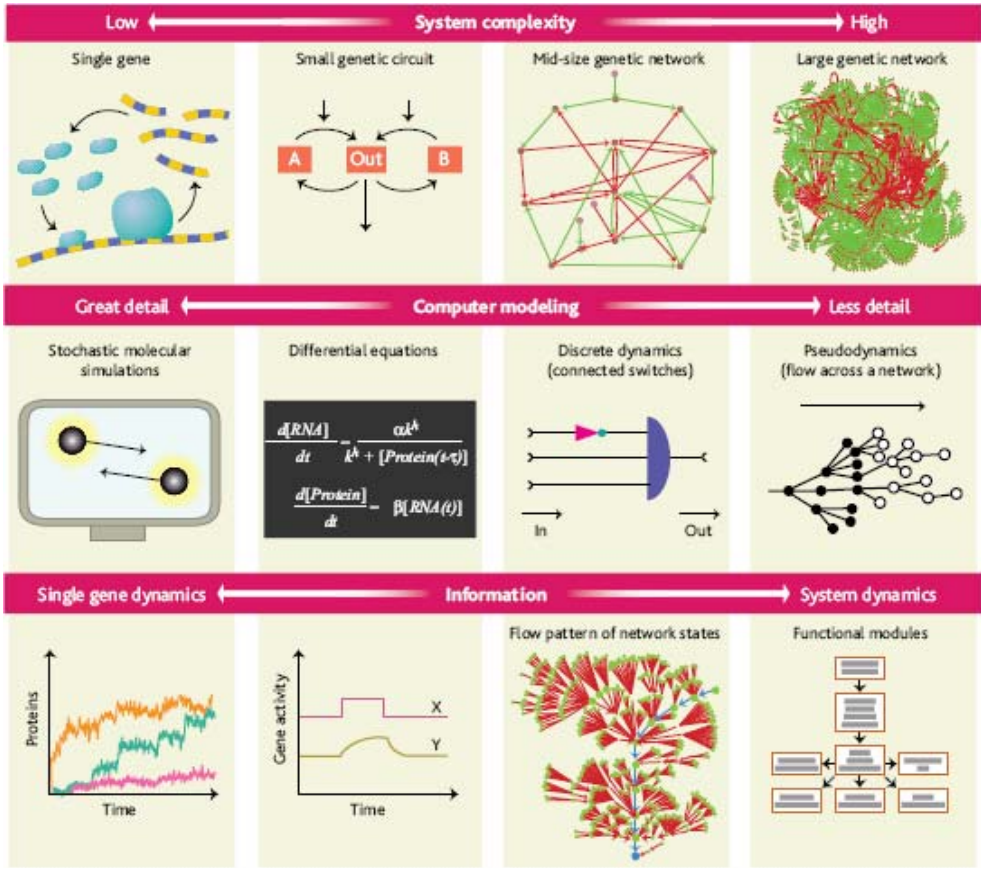


Figure 3.3 The different levels of description in models of genetic networks. Whereas single genes can be modeled in molecular detail with stochastic simulations (Left column), a differential equation representation of gene dynamics is more practical when turning to circuits of genes (center left column). Approximating gene dynamics by switchlike ON/OFF behavior allows modeling of mide-sized genetic circuits (center right column) and still faithfully represents the overall dynamics of the biological system. Large genetic networks are currently out of reach for predictive simulations. However, more simplified dynamics, such as percolating flows across a network structure, can teach us about the functional structure of a large network (right column).
Figure from Bornholdt S., Less is More in Modeling Large Genetic Networks. Science 310, 449-450 (2005).

A first level of coarse-graining in genetic regulation already exists in the standard approach of modeling protein and RNA concentrations with specific equations called “ordinary” differential equations. These equations nicely summarize the molecular interactions that make up the cellular machinery that regulates the activity of a gene. When at least a few tens of molecules are involved in regulating a gene, details of the interactions can usually be neglected, and interaction rates can be used instead of tracking the single molecular binding events [24]. With large networks involving thousands of regulatory genes (genes that encode proteins that regulate other genes), the number of differential equations needed to describe the system can become huge. The sheer number of parameters (such as decay rates, production rates, and interaction strengths) in this mathematical model poses a challenge, both for experiment and theory. A central question is what the right level of description is when constructing quantitative models of large or even system wide genetic networks (see Fig. **3.3**). Is coarse-graining of genetic network models possible? A number of general building blocks identified in genetic

networks at least indicate that robust simplified models are possible. Modules such as autoregulatory excitatory (positive) feedback loops (which can convert a transient signal into a sustained signal and thus serve as “storage” devices), inhibitory feedback loops (which suppress instability due to noise), or feed-forward loops (which may enhance responsiveness of a gene) represent different kinds of robust switching elements. Brandman *et al*. [32] describe another such building block, the dual positive-feedback loop, which is frequently found in subnetworks of larger cellular and genetic networks. But why would cells have evolved two positive feedback loops when one is enough to create a switch? Brandman et al. [32] find that the combination of the two loops can make genetic switching faster and, at the same time, reduce signal noise. A slow loop creates robustness in the signal, whereas a fast loop allows for switching speed. Given the quite complex cellular machinery that is needed to run this dual positive feedback circuit with biochemical means, its dynamic behavior is intriguingly simple. It functions as a particularly robust, yet fast switch that is reminiscent of the robustly designed electronic building blocks used to build modern computers. This observation provides support for discrete models of genetic networks in which genes are modeled as switchlike dynamic elements that are either ON or OFF. The first such models, generated about four decades ago, were random networks of discrete dynamical elements, as few data about regulatory genetic networks were available at that time [33]. These models were long considered to be merely a speculative analogy. However, recent advances in modeling combined with the first opportunities to validate genetic network models with data from living cells show that simplified network models, such as those representing a regulatory gene as a binary switch ON/OFF type, can indeed predict the overall dynamical trajectory of a biological genetic circuit. For example, the trajectory of the segment polarity network in the fly *Drosophila melanogaster* has been predicted solely on the basis of discrete binary model genes [34]. Similarly, a dynamic binary model of the genetic network that controls the yeast cell cycle was constructed in 2004 by Li *et. al* [35]. In both systems, the dynamics converge to so-called attractors (states or sequences of states of the genes) and for these, the models match the biological dynamics. These dynamical attractors seem to depend not so much on the details of the kinetic constants, but more on the circuit wiring. Insensitivity to biochemical kinetic parameters indicates that for understanding the dynamics of these circuits, it is their wiring that is most important [36]. This seems to be the reason why large genetic networks can be represented as networks of discrete dynamic elements, without the tuning of parameters.

Simplified models on even larger scales should be envisioned for real progress in this field. However, modeling of large cellular networks is often hampered by incomplete knowledge of the full circuitry, despite a wealth of data. An example of how simplification of the dynamics of single elements enables us to gain valuable information about a system’s function is presented in the recent article by Ma’ayan *et al*. [37]. They propose a discrete “pseudodynamics” of binary states that percolates through the known part of a 1500-node mammalian cellular network and gives a rough but informative estimate of the regulatory information flow through the system. The thousands of parameters required to generate a standard differential equations model of all the relevant biochemical interactions has been neglected by the others in favor

of a statistical perspective that provides valuable information about the global architecture of a cellular network. It is not a direct representation of the biochemical dynamics and does not allow a detailed dynamic simulation of the network. However, it is an analog of the potential propagation of a signal and therefore useful to determine the global signalling structure of an overall network. This approach is error tolerant and gives a robust picture of the overall global modular structure of a network. The simple dynamics of the building blocks points to an interesting perspective for our further understanding of genetic networks. Distinguishing between the robust effective dynamics of a genetic or regulatory switch and the biochemical that are use to practically run it shows that, to understand the system, we do not have to pass trough all the details of the biochemistry. Characterizing the topology of the circuit seems to be the most important consideration, and when going “dynamic,” a clever way to throw away details may be the most important part of model building.

3.3 Gene expression clustering

Clustering is often one of the first steps in gene expression analysis. Our ability to gather genome-wide expression data has far outstripped the ability of our feeble human brains to process the raw data. We can distill the data down to a more comprehensible level by subdividing the genes into a smaller number of categories and then analyzing those. This is where clustering comes in.

The goal of clustering is to subdivide a set of items (in our case, genes) in such a way that similar items fall into the same cluster, whereas dissimilar items fall in different clusters. This brings up two questions: first, how do we decide what is similar; and second, how do we use this to cluster the items? The fact that these two questions can often be answered independently contributes to the bewildering variety of clustering algorithms [38]. Gene expression clustering allows an open exploration of the data, without getting lost among the thousands of individual genes. Beyond simple visualization, there are also some important computational applications for gene clusters. For example, Tavazoie *et al.* [39] used clustering to identify *cis*-regulatory sequences in the promoters of tightly coexpressed genes. Gene expression clusters also tend to be significantly enriched for specific functional categories, which may be used to infer a functional role for unknown genes in the same cluster.

In this section, we focus specially on clustering genes that show similar expression patterns across a number of samples, rather than clustering the samples themselves (or both). The goal is to leave the reader with some understanding of clustering in general and three of the more popular algorithms in particular. Where possible, an attempt is made to provide some practical guidelines for applying cluster analysis to any other gene expression data sets.

It is easy to invent yet another clustering algorithm. There are hundreds of published clustering algorithms, dozens of which have been applied to gene expression data. It is much harder to do a fair evaluation of how well a new algorithm will perform on typical expression data sets, how it compares with

those dozens of other published algorithms and under which circumstances one algorithm should be preferred over another.

There is no one-size-fits-all solution to clustering, or even a consensus of what a "good" clustering should look like. In the words of Jain and Dubes [40]: *"There is no single best criterion for obtaining a partition because no precise and workable definition of cluster exists. Clusters can be of any arbitrary shapes and sizes in a multidimensional pattern space. Each clustering criterion imposes a certain structure on the data, and if the data happen to conform to the requirements of a particular criterion, the true clusters are recovered"*. In other words, each algorithm imposes its own set of biases on the clusters it constructs, and whereas most sensible clustering algorithms may yield similar results on trivial test problems, in practice they can give widely differing results on messy real world expression data. So, how do we decide how similar the expression patterns of two genes are? Note that this really boils down to which types of expression patterns we would like to see fall into the same clusters, something that may go well beyond which patterns look visually similar and is directly related to the question "*what do we want to achieve by clustering?*".

Two of the easiest and most commonly used similarity measures for gene expression data are Euclidean distance and Pearson correlation coefficient (See Table **1.3** for these and other similarity measures and variants). Note in the case of the Euclidean distance notice that it is sensitive to scaling and difference in average expression level, whereas correlation is not.

The two most important classes of clustering methods are hierarchical clustering and partitioning (Fig. **4.3**). In hierarchical clustering, each cluster is subdivided into smaller clusters, forming a tree-shaped data structure or dendrogram. Agglomerative hierarchical clustering (also used in phylogenetics) starts with the single–gene clusters and successively joins the closest clusters until all genes have been joined into the supercluster. In fact, there is a whole family of clustering methods, differing only in the way intercluster distance is defined. Some of the more common ones are single linkage, complete linkage, average linkage and centroid linkage.

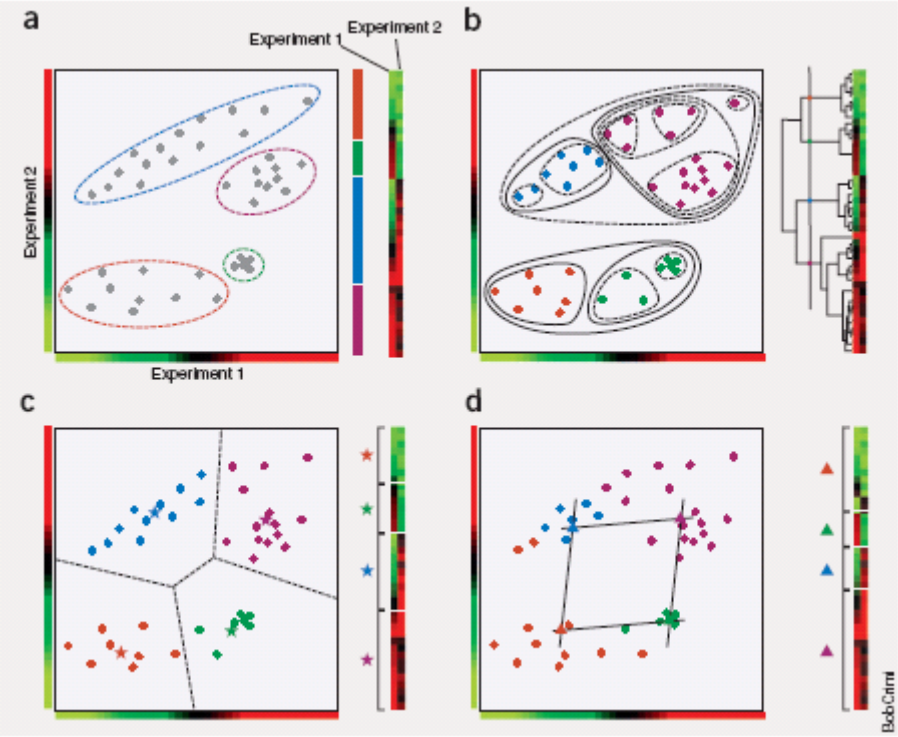


Figure 4.3 A simple clustering example with 40 genes measured under two different conditions. **(a)** The data set contains four clusters of different sizes, shapes and number of genes. Left: each dot represents a gene, plotted against its expression value under the two experimental conditions. Euclidean distance, which corresponds to the straight-line distance between points in this graph, was used for clustering. Right: the standard red-green representation of the data and corresponding cluster identities. **(b)** Hierarchical clustering finds an entire hierarchy of clusters. The tree was cut at the level indicated to yield four clusters. Some of the superclusters and subclusters are illustrated on the left. **(c)** k-means (with k=4) partitions the space into four cluster centroids (stars) is closest. **(d)** So-called organized map technique (SOM) finds clusters, which are organized into a grid structure (in this case a simple 2 X 2 grid).
Figure from D`haeseleer P., How does gene expression clustering work? Nature Biotechnology 23, 1499-1501 (2005).

Partitioning methods, on the other hand subdivided the data into a typically predetermined number of subsets, without any implied hierarchical relationship between these clusters. How many clusters are actually present in the data is a thorny issue. A common approach is to rerun the clustering with different number of clusters, in the hope of being able to distinguish the optimal number of clusters. A hierarchical clustering can also be reduced to a partitioning, by cutting the dendrogram at a given level (Fig. **4.3b**).

The quality of a clustering result can be evaluated by means of internal criteria (that is, based on various statistical properties of the clusters) or external criteria (that is, based on additional information that was not used in the clustering process itself). Internal validation seems straightforward: we would like clusters to be compact and well separated. Unfortunately, this reverts back to the same tricky question "*what would we like clusters to look like?* At least a dozen different measures have been developed to test the quality of a cluster, and for many of these there exists a clustering method that will optimize that measure. For example, k-means optimizes the variance of the clusters, whereas complete

linkage minimizes the radius of the clusters. Other measures test the within-group versus between-group variance, the separation between clusters and the stability of clusters with respect to noise, random initializations (such as for k-means or SOM) and leaving out other conditions, see for example [41-44].

Table 1 Gene expression similarity measures

Manhattan distance (city-block distance, L1 norm)	$d_{fg} = \sum_c \lvert e_{fc} - e_{gc} \rvert$
Euclidean distance (L2 norm)	$d_{fg} = \sqrt{\sum_c (e_{fc} - e_{gc})^2}$
Mahalanobis distance	$d_{fg} = (\mathbf{e}_f - \mathbf{e}_g)^t \Sigma^{-1} (\mathbf{e}_f - \mathbf{e}_g)$, where Σ is the (full or within-cluster) covariance matrix of the data
Pearson correlation (centered correlation)	$d_{fg} = 1 - r_{fg}$, with $r_{fg} = \frac{\sum_c (e_{fc} - \bar{e}_f)(e_{gc} - \bar{e}_g)}{\sqrt{\sum_c (e_{fc} - \bar{e}_f)^2 \sum_c (e_{gc} - \bar{e}_g)^2}}$
Uncentered correlation (angular separation, cosine angle)	$d_{fg} = 1 - r_{fg}$, with $r_{fg} = \frac{\sum_c e_{fc} e_{gc}}{\sqrt{\sum_c e_{fc}^2 \sum_c e_{gc}^2}}$
Spellman rank correlation	As Pearson correlation, but replace e_{gc} with the rank of e_{gc} within the expression values of gene g across all conditions $c = 1 \ldots C$
Absolute or squared correlation	$d_{fg} = 1 - \lvert r_{fg} \rvert$ or $d_{fg} = 1 - r_{fg}^2$

d_{fg}, distance between expression patterns for genes f and g. e_{gc}, expression level of gene g under condition c.

Table 1.3 Gene expression similarity measures.

Ultimately, the real test of any clustering procedure is he coincidence with the biological facts. The most reliable quality measure of a clustering method is how well it actually performs the task at hand. For example, if our goal is to cluster together genes with similar function, then we can use existing functional annotations to verify how well the goal has been achieved [42, 43]. If our goal is to extract *cis*-regulatory elements from the clusters, then we can check how well genes with known regulatory sequences are clustered together.

3.4 The Potts model

Ferromagnetic Potts models have been studied extensively for many years in statistical physics [45]. The basic spin variable s can take one of q integer values: s = 1, 2… q. In a magnetic model the Potts spins are located at points v_i that usually reside on the sites of a finite lattice made of N sites. Pairs of spins associated with points i and j are coupled by an interaction of strength $J_{ij} > 0$. Denote by S a configuration of the system, $S = \{s_i\}_{i=1}^N$. The energy of such a configuration is given by the Hamiltonian

$$H(s) = \sum_{\langle i,j \rangle} J_{ij}(1-\delta_{s_i,s_j}) \quad s_i = 1,\ldots,q, \tag{3.1}$$

where the notation $\langle i, j \rangle$ stands for neighboring sites v_i *and* v_j. The contribution of a pair $\langle i, j \rangle$ to H is 0 when $s_i = s_j$, that is, when the two spins are aligned, and is $J_{ij} > 0$ otherwise. If one chooses interactions that are a decreasing function of the distance $d_{ij} \equiv d(v_i, v_j)$, then the closer two points are to each other, the more they "like" to be in the same state. The Potts Hamiltonian is very similar to other energy functions used in neural systems, where each spin represents a q-state neuron with an excitatory coupling to its neighbors. In fact, magnetic models have inspired many neural models [46]. In order to calculate the thermodynamics average of a physical quantity *A* at a fixed temperature *T*, one has to calculate the sum

$$\langle A \rangle = \sum_S A(s)P(s) \ , \tag{3.2}$$

where *P(s)* is the Boltzmann factor,

$$P(s) = \frac{1}{Z}\exp\left(-\frac{H(s)}{T}\right). \tag{3.3}$$

The latter plays the role of the probability density, which gives the statistical weight of each spin configuration $S = \{s_i\}_{i=1}^N$ in thermal equilibrium and *Z* is a normalization constant, $Z = \Sigma_S \exp(-H(s)/T)$.

Some of the most important physical quantities (*A*) for this magnetic system are the order parameter or magnetization and the set of $\delta_{si,sj}$ functions, because their thermal averages reflect the ordering properties of the model.

The order parameter of the system is $\langle m \rangle$, where the magnetization, *m(s),* associated with a spin configuration *S* is defined as [47]

$$m(S) = \frac{qN_{\max}(S) - N}{(q-1)N} \tag{3.4}$$

with $N_{max}(S) = \max \{ N_1(S), N_2(S), \ldots N_q(S)\}$, where $N_\mu(s)$ is the number of spins with the value μ; $N_\mu(s) = \Sigma_i \, \delta_{s_i,\mu}$.

The thermal average of δ_{s_i,s_j} is called the spin-spin correlation function,

$$G_{ij} = \langle \delta_{s_i,s_j} \rangle, \tag{3.5}$$

which is the probability of the two spins s_i and s_j to be aligned.

The case $J_{ij} = J$

Paramagnetic phase

When the spins are on the lattice and all nearest-neighbor couplings are equal, $J_{ij} = J$, the Potts system is homogeneous. Such a model exhibits two phases. At high temperatures, the system is paramagnetic or disordered, $\langle m \rangle = 0$, indicating that $N_{max}(s) \approx N/q$ for all statistically significant configurations. In this phase the correlation function G_{ij} decays to $1/q$ when the distance between points v_i *and* v_j is large; then 1/q is the probability of finding two completely independent Potts spins in the same state. At very high temperatures, even neighboring sites have $G_{ij} \approx 1/q$.

Ferromagnetic phase

As the temperature is lowered, the system undergoes a sharp transition to an ordered, ferromagnetic phase; the magnetization jumps to $\langle m \rangle \neq 0$. This means that in the physically relevant configurations (at low temperatures), one Potts state "dominates" and $N_{max}(s)$ exceeds N/q by a macroscopic number of sites. At very low temperatures, $N_{max}(s) \approx N$ and therefore $\langle m \rangle \approx 1$ and $G_{ij} \approx 1$ for all pairs $\{v_i , v_j\}$.

The variance of the magnetization is related to a relevant thermal quantity, the susceptibility,

$$\chi = \frac{N}{T}\left(\left\langle m^2 \right\rangle - \left\langle m \right\rangle^2\right), \tag{3.6}$$

which also reflects the thermodynamic phases of the system. At low temperatures, fluctuations of the magnetization are negligible, so the susceptibility χ is small in the ferromagnetic phase.

The case of unequal *J_{ij}: the additional superparamagnetic phase*

This is the case of (strongly) inhomogeneous Potts models.The connection between Potts spins and clusters of aligned spins in this situation was established by Fortuin and Kasteleyn in 1972 [48]. The inhomogeneous models describe the more complicated cases when the spins form magnetic "grains", with very strong couplings between neighbors that belong to the same grain and very weak interactions between all other pairs. At low temperatures, such a system is also ferromagnetic, but as the temperature is raised, the system may exhibit an intermediate, *superparamagnetic phase*. In this phase strongly coupled grains are aligned (that is, are in their respective ferromagnetic phases), while there is no relative ordering of different grains (that is, a globally paramagnetic phase).

The range of the superparamagnetic phase

At the transition temperature from the ferromagnetic to superparamagnetic phase a pronounced peak in the plot of χ vs. T is observed [49]. In the superparamagnetic phase, fluctuations of the state taken by grains acting as a whole (that is, as giant superspins) produce large fluctuations in the magnetization. As the temperature is raised further, the superparamagnetic to paramagnetic transition is reached; each grain disorders, and χ abruptly diminishes by a factor that is roughly proportional to the the size of the largest cluster. Thus, the temperatures where a peak of the susceptibility occurs and the temperatures at which χ decreases abruptly provide the range of temperatures in which the system is in its superparamagnetic phase.

In principle, one can have a sequence of several transitions in the superparamagnetic phase. As the temperature is raised, the system may break first into two clusters, each of which breaks into more (still macroscopic) subclusters, and so on. Such an evolutionary structure of the magnetic clusters reflects a hierarchical organization of the data into categories and subcategories. To gain some analytic insight into the behavior of inhomogeneous Potts ferromagnets, we calculated the properties of such a "granular" system with a macroscopic number of bonds for each spin. For such ("infinite-range") models, the mean field approach is exact [50]. In the paramagnetic phase, the spin state at each site is independent of any other spin, that is, G_{ij} =1/q.

At the paramagnetic-superparamagnetic transition the correlation between spins belonging to the same group jumps abruptly to

$$G_{i,j}^{p\to sp} = \frac{q-1}{q}\left(\frac{q-2}{q-1}\right)^2 + \frac{1}{q} \cong 1 - \frac{2}{q} + O\left(\frac{1}{q^2}\right), \tag{3.7}$$

while the correlation between spins belonging to different groups is unchanged. On the other hand, the ferromagnetic phase is characterized by strong correlations between all spins of the system:

$$G_{i,j}^{ferro} \geq \frac{q-1}{q}\left(\frac{q-2}{q-1}\right)^2 + \frac{1}{q}. \tag{3.8}$$

There is an important lesson to remember from this: in mean field we see that in the superparamagnetic phase, two spins that belong to the same grain are strongly correlated, whereas for pairs that do not belong to the same grain, $G_{i,j}$ is small. As it turns out, this double-peaked distribution of the correlations is not an artifact of mean field and will be used in our solution of the problem of data clustering.

References

[1] Kanehisa, M., Post-Genome Informatics, Oxford University Press, Okford (2000).

[2] Kanehisa, M., and Goto, S., KEGG: Kyoto encyclopedia of genes and genomes. *Nucl. Acids Res*. **28(1)**, 27-30 (2000).

[3] Brownstein M.J., Trent J.M. and Boguski M.S. Functional genomics. In M. Patterson and M. Handel, eds. *Trends Guide to Bioinformatics*, 27-29, Elsevier, Oxford (1998)

[4] Collado-Vides J., A transformational-grammar approach to the study of the regulation of gene expression. *J. Theor. Biol*. **136**, 403-425 (1989).

[5] Collado-Vides J., Integrative representations of the regulation of gene expression. In J. Collado-Vides, B. Magasanik, and T. F. Smith eds. *Integrative Approaches to Molecular Biology*, 179-203, MIT Press, Cambridge, M.A (1996).

[6] Collado-Vides J., Gutierrez-Ríos R.M. and Bel-Enguix G., Networks of transcriptional regulation encoged in a gramatical model. *BioSystems* **47**, 103-118 (1998).

[7] Collado-Vides J., Magasanik B., and Smith, T.F., eds. *Integrative Approaches to Molecular Biology*, MIT Press, Cambridge, MA. (1996).

[8] Palsson B. O., What lies beyond bioinformatics? *Nat. Biotechnol*. **15**, 3-4 (1997).

[9] Strohman R.C., The coming Kuhnian revolution in biology. *Nat. Biotechnol*. **15**, 194-200 (1997).

[10] Thieffry D., From global expression data to gene networks. *BioEssays* **21** (11), 895-899 (1999).

[11] Thieffry D., Colet M. and Thomas, R., Formalisation of regulatory nets: A logical method and its automatization. *Math. Modelling Sci. Computing* **2**, 144-151 (1993).

[12] Thieffry D., Huerta A.M., Pérez-Rueda E. and Collado-Vides J., From specific gene regulation to genomic networks: A global analysis of transcriptional regulation in Escherichia coli. *BioEssays* **20**, 433-440 (1998).

[13] Thieffry D. and Romero D., The modularity of biological regulatory networks, *BioSystems* **50**, 49-59 (1999).

[14] Brown P.A. and Botstein D., Exploring the new world of the genome with DNA microarrays. *Nature Genet*. **21** (suppl.), 33-37 (1999).

[15] Brown M., Grundy W., Lin N., Cristianini D., Sugnet N., Furey C., Ares T. and Haussler D., Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc.Natl. Acad. Scie. USA* **97(1)**, 262-267 (2000).

[16] Lipschutz R.J., Fodor S.P.A., Gingeras T.R., and Lockhart D.J., High density synthetic oligonucleotide arrays. *Nat Genet*. **21** (suppl.), 20-24 (1999).

[17] Lockhart D.J., and Winzeler E.A., Genomics, gene expression and DNA arrays. *Nature* **405**, 827-836 (2000).

[18] Kahn P., From genome to proteome: Looking at cell´s protein. *Science* **270**, 369-370 (1995).

[19] Mann, M., Quantitative proteomics. *Nat. Biotechnol*. **17**, 954-955 (1999).

[20] Pandey A. and M., Proteomics to study genes and genomes. Nature **405**, 837-846 (2000).

[21] Zhu H. and Snyder M., Protein arrays and microarrays. *Curr.Opin.chem.Biol*. **5**, 40-45 (2001).

[22] Endy D. and Brent R., Modelling cellular behavior. *Nature* **409**, 391-395 (2001).

[23] Hasty, J., McMillen, D., Isaacs, F., and Collins, J.J., Computational studies of gene regulatory networks: In numero molecular biology. *Nat. Rev. Genet*. **2**, 268-279 (2001).

[24] McAdams H.M. and Arkin A., Stochastic mechanisms in gene expression. *Proc.Natl.Acad. Sci. USA* **94**, 814-819 (1997).

[25] McAdams H.H. and Arkin A., Simulation of prokaryotic genetic circuits. *Ann.Rev.Biophys. Biomol. Struct*. **27**, 199-224 (1999).

[26] McAdams H.H. and Arkin A., It´s a noisy business ! Genetic regulation at the nanomolar scale. *Trends Genet*. **15(2)**, 65-69 (1999).

[27] McAdams H.H. and Shapiro L., Circuit simulation of genetic networks. Science **269**, 650-656 (1995).

[28] Smolen P., Baxter D.A. and Byrne J.H., Effects of macromolecular transport ans stochastic fluctuations on dynamics of genetic regulatory systems. *Am.J. Phys*. **277**, C777-C790 (1999).

[29] Smolen P., Baxter D.A. and Byrne J.H., Modeling transcriptional control in gene networks: Methods, recent results, and future directions. *Bull. Math. Biol*. **62**, 247-292 (2000).

[30] Bower J.M. and Bolouri H., Computational Modeling of Genetic and Biochemical Networks, MIT, Press, Cambridge, MA (2001).

[31] Bornholdt, S., Less is more in modeling large genetic networks. *Science*. **310**, 449-451 (2005).

[32] Brandman O., J. E. Ferrell Jr., R. Li,T.Meyer, Interlinked fast and slow positive feedback loops drive reliable cell decisions. *Science* **310**, 496 (2005).

[33] Kauffman S. A., Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol*. **22**, 437-467 (1969).

[34] Albert R., Othmer H.G., The topology of the regulatory interactions predict the expression pattern of the segment polarity genes in Drosophila melanogaster, *J. Theor. Biol*. **223**, 1-18 (2003).

[35] Li F., Long T., Lu Y., Ouyang Q., Tang C., The yeast cell-cycle network is robustly designed circuit topology and the evolution of robustness in two-gene circadian oscillators. *Proc. Natl. Acad. Sci. U.S.A*. **101**, 4781-4786 (2004).

[36] Wagner A., *Proc. Natl. Acad. Sci. U.S.A*. **102**, 11775 (2005).

[37] Ma'ayan A. et al., Formation of regulatory patterns during signal propagation in a mammalian cellular network. *Science* **309**, 1078-1083 (2005).

[38] D´haeseleer P., How does gene expression clustering work? *Nature Biotechnology* **23**, 1499-1501 (2005).

[39] Tavazoie S., Hughes J.D., Campbell M.J., Cho R.J. & Church G.M., Systematic determination of genetic network architecture. *Nature Genetics* **22**, 281-285 (1999).

[40] Jain A.K. & Dubes R.C., Algorithms for clustering data. (Prentice-Hall, Englewood Cliffs, New Jersey, 1988).

[41] Handl J., Knowles J. & Kell D.B., Computational cluster validation in post-genomic data analysis. *Bioinformatics* **21**, 3201-3212 (2005).

[42] Gibbons F.D. & Roth F.P., Judging the quality of gene expression-based clustering methods using gene annotation. *Genome Res*. **12**, 1574-1581 (2002).

[43] Costa I.G., de Carvalho F.A. & de Souto M.C., Comparative analysis of clustering methods for gene expression time course data. *Genet. Mol. Biol*. **27**, 633-639 (2004).

[44] Datta S. & Datta S., Comparisons and validation of statistical clustering techniques for microarray gene expression data. *Bioinformatics* **19**, 459-466 (2003).

[45] Wu F.Y., The Potts model. *Reviews of Modern Physics* **54**(1), 235-268 (1982).

[46] Hertz J., Krogh A. and Palmer R., Introduction to the theory of neural computation, Redwood City, CA: Addison-Wesley (1991).

[47] Chen S., Ferrenberg A.M. and Landau D.P., Randomness-induced second order transitions in the two dimensional eight state Potts model: A Monte Carlo study. *Physical Review Letters*, **69** (8), 1213-1215 (1992).

[48] Fortuin C.M. and Kasteleyn P.W., On the random-cluster model. *Physica (Urecht)*, **57**, 536-564 (1972).

[49] Blatt M., Wiseman S., Domany E. Superparamagnetic clustering of data. *Physical Review Letters*, **76**, 3251-3255 (1996).

[50] Wiseman S., Blatt M. and Domany E. Unpublished manuscript.

Chapter 4
Maximum Entropy in the Gene Expression

DNA microarrays allow us to explore a major subset or all genes of an organism. In the case of the metabolic architecture for less complex organisms, such as Escherichia coli, the biochemical network has been described in much detail. Here, we analyze the clustering of such networks by applying gene expression data that define edge lengths in the network. As a novel idea, we use the concept of entropy in a fundamental condition that all clustering algorithm must satisfy. This entropy condition is basically related to the microarray data analysis as a sort of thermodinamical equilibrium condition. In addition, we investigate the clustering of such networks by applying the Potts spin model to the gene expression data by introducing an interaction between neighboring spins, whose strength is a decreasing function of the distance between neighbors. We tested our method on gene expression data from E. coli and we notice different results with respect to the published paper although it fits well to the commonly knowns regulatory netwoks of the cell biological processes.

4.1 Introduction

DNA microarray is one of the latest breakthroughs in experimental molecular biology. This technology permits the analysis of gene expression, DNA sequence variation, protein levels, tissues, cells and other chemicals structures in massive format [1, 2]. However, the analysis and handling of such fast growing data is becoming a major challenge in the utilization of the technology. Powerful mathematical and statistical methods are therefore called for this purpose to search for orderly features and logical relationships in this type of data.

Over the last 3 decades, biochemical investigations led to the discovery of a self consistent picture of the metabolism of the cell metabolism [3]. In fact, for

less complex organisms, like *Echerichia coli*, the metabolic network has been almost completely described [4]. For yeast, knowledge derived from such biochemical networks was used to support the clustering paradigm of gene expression data [5].

Extraction of meaningful information from gene expression data is a complex task because of the large volume of data and the expected complexity of its structure and organization. To address this problem in an unbiased way several specialized methods, broadly known as “clustering techniques,” have been developed recently [6-8]. All these techniques, though differing in details, in essence try to identify genes (or samples) that behave similarly across the samples (or genes) and classify them as belonging to one group or a definite cluster. Among the clustering techniques that employ the concepts of physics, the one that succeeds in correctly clustering most of the types of data [9] is the “superparmagnetic” clustering method that has been promoted by Domany and coworkers about a decade ago [7]. This method exploits the propieties of phase transitions in disordered Potts ferromagnets.

The goal of clustering in gene expression processes is to subdivide a set of genes in such a way that similar genes fall into the same cluster, whereas dissimilar genes fall in different clusters. However, how do we decide what is similar and how do we use this to cluster the genes. Although many of the proposed algorithms have been reported to be successful, no single algorithm has emerged as a method of definitive choice. Further, the issues of determining the “correct” number of clusters and the choice of “the best” algorithm are not yet clear [10].

In this chapter, we propose the maximum entropy condition of determining the correct way to cluster the genes. In Physics, the entropy is a measure of the level of constraint or order that exists so that a process can be carried out. In particular, it has been shown that the progressive tendency of system to go away from equilibrium is governed by a law of maximum entropy production [11]. In concrete terms, we apply the Potts spin clustering technique to cluster gene expression data by using the information about nearest neighbour relations of biochemical networks. Only clusters of neighbouring genes with similar expression profiles could occur in real situation. Here the maximum entropy condition is applied to the gene expression data of *E. coli* provided by Khodursky *et al*. [12].

4.2 Super-paramagnetic clustering

Super-paramagnetic clustering is a hierarchical method, based on the Potts model of magnetic spins. The algorithm assigns to each spin a data point. The spins are correlated to each other and this correlation is reflected in the energy of the system, which is minimal when all the spins are aligned and maximal when each spin points to a different direction with respect to all the other spins of the system. The parameter that we denote by T corresponds to the temperature parameter in the statistical mechanism of the physical Potts model. However, in the clustering procedure, it controls the clustering resolution, namely the cluster divisibility of the system. At low values of this resolution parameter, all the spins are aligned and form a single cluster. That means that it

does not matter if we see a single spin or the hold cluster because the orientation is the same. This is equivalent to the *ferromagnetic* phase of the physical system. As the control parameter increases, the gene expression system undergoes a sequence of phase transitions. By measuring spin-spin correlations at each value of T we determine the probability that two data points share the same spin orientation and if this probability is high, the pair of corresponding data points are placed in the same cluster. At very high temperature values, the orientation of the spins are uncorrelated and each data point becomes independent with respect to the other data points, as if it makes a single cluster. This corresponds to the *paramagnetic* phase of the physical system, where correlations are short ranged. On the other hand, the *super-paramagnetic* phase is intermediate between the previous two phases, corresponding to a phase in which large sub-clusters of the cluster of all data points may occur. In this phase, the data points form highly correlated domains in which a data point in one domain is uncorrelated with data points belonging to the other domains.

4.3 Monte Carlo simulation of Potts models: the Swendsen-Wang method

The aim of equilibrium statistical mechanics is to evaluate sums such as equation 3.2 for models with $N >> 1$ spins. This can be done analytically only for very limited cases. One resorts therefore to various approximations or to computer simulations that aim at evaluating thermal averages numerically.

Direct evaluation of sums like equation 3.2 is impractical, since the number of configurations S increases exponentially with the system size *N*. Monte Carlo simulation methods overcome this problem by generating a characteristic subset of configurations, which are used as a statistical sample [13]. They are based on the notion of *importance sampling*, in which a set of spin configurations $\{S_1, S_2, \ldots, S_M\}$ is generated according to the Boltzmann probability distribution (see equation 3.3). Then, expression 3.2 is reduced to a simple arithmetic average,

$$\langle A \rangle \approx \frac{1}{M} \sum_{i}^{M} A(s_i), \qquad (4.1)$$

where the number of configurations in the sample, *M*, is much smaller than q^N, the total number of configurations. The set of *M* states necessary for the implementation of equation 4.1 is constructed by means of a Markov process in the configuration space of the system. There are many ways to generate such a Markov chain. A very efficient one is the Swendsen-Wang Monte Carlo procedure [14,15]. The main reason for choosing the latter procedure is that it is perfectly suitable for working in the superparamagnetic phase: it overturns an aligned cluster in only one Monte Carlo step, whereas algorithms that use standard local moves will take forever to do this. The steps of the SW algorithms are the following:

First step

The first configuration can be chosen at random (or by setting all $s_i = 1$). Thus, we already generated n configurations of the system, $\{S_i\}_{i=1}^{n}$, and we start to generate configuration $n+1$.

Second step

Visit all pairs of spins $\langle i, j \rangle$ that interact, that is, having $J_{i,j} > 0$; the two spins are frozen together with probability

$$P_{i,j}^{f} = 1 - \exp\left(-\frac{J_{i,j}}{T} \delta_{s_i, s_j} \right). \tag{4.2}$$

That is, if in our current configuration S_n the two spins are in the same state, $s_i = s_j$, then sites i and j are maintained unchanged (frozen) with probability $p^f = 1 - \exp(-J_{ij}/T)$.

Third Step

Having gone over all the interacting pairs, the next step of the algorithm is the task to *identify the SW clusters of spins*. By definition an SW cluster contains all spins that have a path of frozen bonds connecting them. Note that according to equation 4.2, only spins of the same value can be frozen in the same SW cluster. After this identification, our N sites are assigned to some number of distinct SW clusters. If we think of the N sites as vertices of a graph whose edges are the interactions between neighbors $J_{i,j} > 0$, each SW cluster is a subgraph of vertices connected by frozen bonds.

Fourth step

The final step of the procedure is to generate the new spin configuration S_{n+1}. This is done by drawing, independently for each SW cluster, randomly, a value $s = 1,\ldots,q$, which is assigned to all its spins, that is, to each SW cluster a certain orientation is assigned. This ends up one Monte Carlo step $S_n \rightarrow S_{n+1}$ in the SW procedure.

By iterating this steps M times while calculating at each Monte Carlo step the physical quantity $A(s_i)$, one can obtain the thermodynamic average $\langle A \rangle$ of equation 4.1. The physical quantities that we are interested in are the magnetization (equation 4.3) and its square value which enter the calculation of the susceptibility χ given in equation 3.6, and the spin-spin correlation function $G_{i,j}$ (equation 3.5). Actually, in most simulations a number of the early configurations are discarded, to allow the system to "forget" its initial state. This is not necessary if the number of configurations M is not too small (increasing M improves the statistical accuracy of the Monte Carlo experiment). Measuring autocorrelation times provides a way of both deciding on the number of discarded configurations and checking that the number of configurations M

generated is sufficiently large [16]. A less rigorous way is simply plotting the energy as a function of the number of SW steps and verifying that the energy reached a stable regime. At temperatures where large regions of correlated spins occur, local methods (such as Metropolis), which flip one spin at a time, become very slow. Since the SW procedure by flips large clusters of aligned spins simultaneously, it exhibits much smaller autocorrelation times than local methods. The efficiency of the SW method, has been all ready tested in various Potts and Ising models [17,18].

4.4 Maximum entropy algorithm

So far we have defined the Potts model, the various "thermodynamic" functions that one measures for this model, and the (numerical) method used to evaluate these quantities. We can now turn directly to the problem for which these concepts will be use: clustering of gene expression data. For the sake of concreteness, assume that our data consist of N patterns of measurements v_i, specified by N corresponding vectors $\mathbf{x}_i$, embedded in a D-dimensional metric space. The starting point is the specification of the Hamiltonian (equation 3.1), which is the assumed to govern the interaction (correlation) of the system. Next, by measuring the susceptibility and magnetization as a functions of the temperature (resolution parameter), the different correlated phases of the interaction model are identified. Finally, the correlation of neighboring pairs of spins, $G_{i,j}$, is determined. This correlation function is then used to partition the spins and the corresponding data points into clusters. The outline of the three stages and the subtasks contained in each can be summarized as follows:

1. Construct the physical analog Potts spin problem. This means:

(a) Associate a Potts spin variable $s_i = 1, 2,\ldots,q$ to each point v_i.
(b) Identify the neighbors of each point v_i according to a selection criterion.
(c) Calculate the interaction $J_{i,j}$ between neighboring points v_i and v_j.

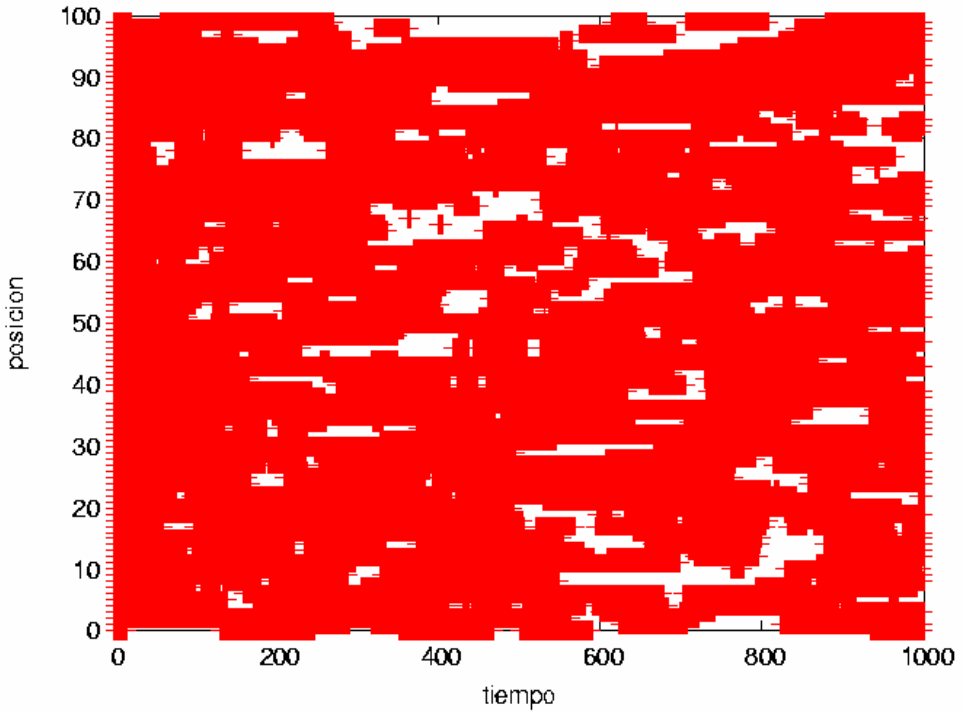
2. Locate the superparamagnetic phase. This requires:

(a) Estimate the (thermal) average magnetization, $\langle m \rangle$, for different temperatures.
(b) Use the susceptibility graph to identify the superparamagnetic phase according to the prescription given on page 27, i.e., between the position of the peak of the susceptibility graph and the point where χ decrease abruptly.

3. Finally, in the superparamagnetic regime, proceed with:

(a) Determining the spin-spin correlation, $G_{i,j}$, for all neighboring points v_i, v_j.
(b) Building the data clusters.

4.5 Results



$$|\alpha_j\rangle = \{\pm\frac{1}{2},\pm\frac{1}{2},...\}$$

$$P(\alpha_j) = \frac{1}{Z}e^{-E(\alpha_j)/kT}$$

Fig 1.4 Random distribution of up and down Ising states.

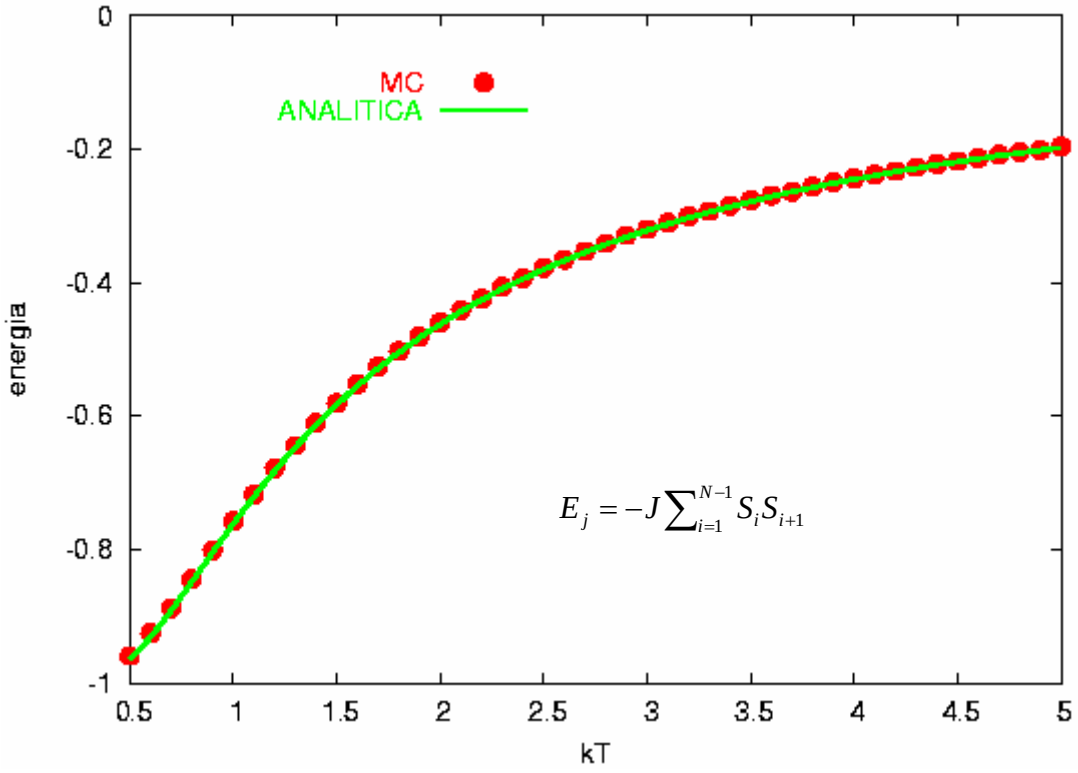


Fig 2.4 The Ising algorithm can simulate the fluctuations around the minimal energy for the α states**.**

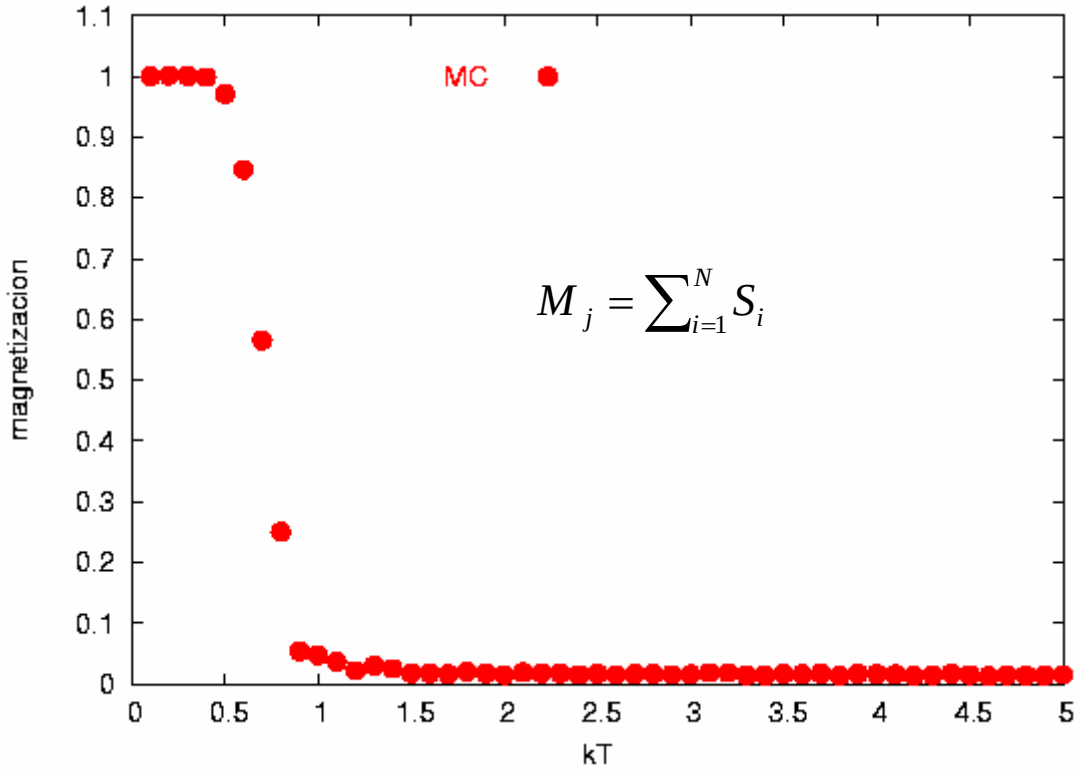


Fig 3.4 For high temperatures, the behavior of spins is random therefore the magnetization is zero. For low temperatures, the magnetization is big.

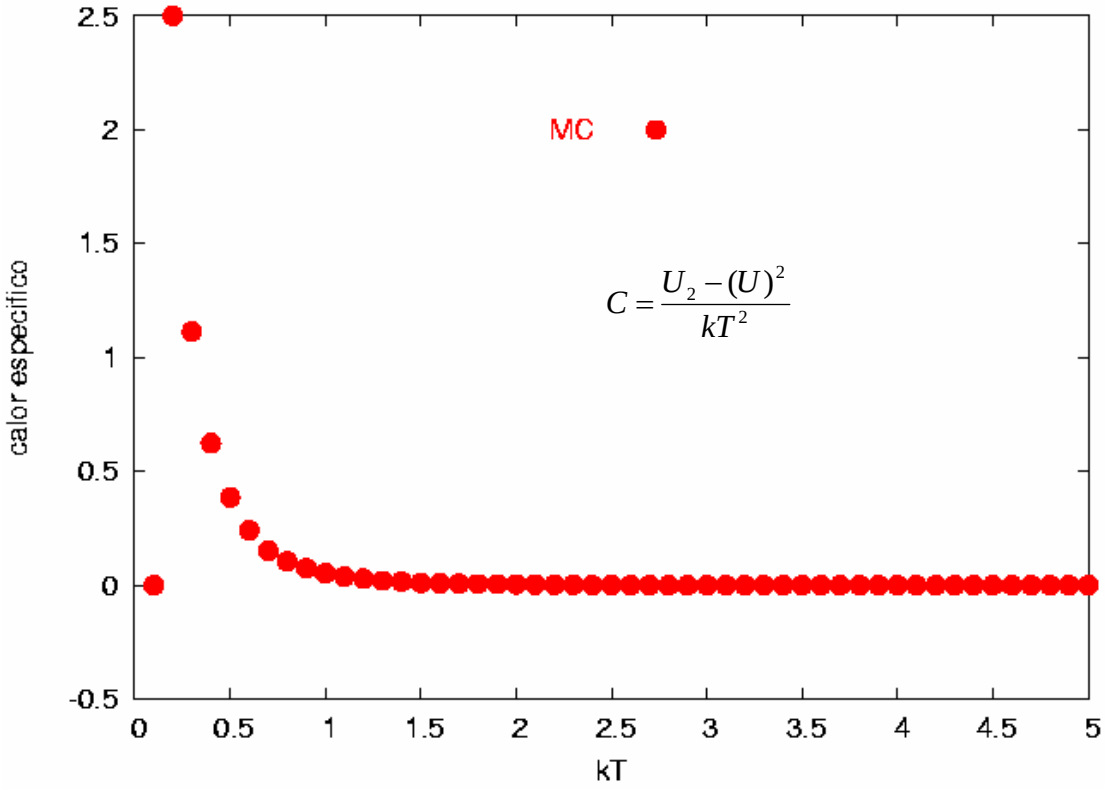


Fig 4.4 The specific heat MC measurement.

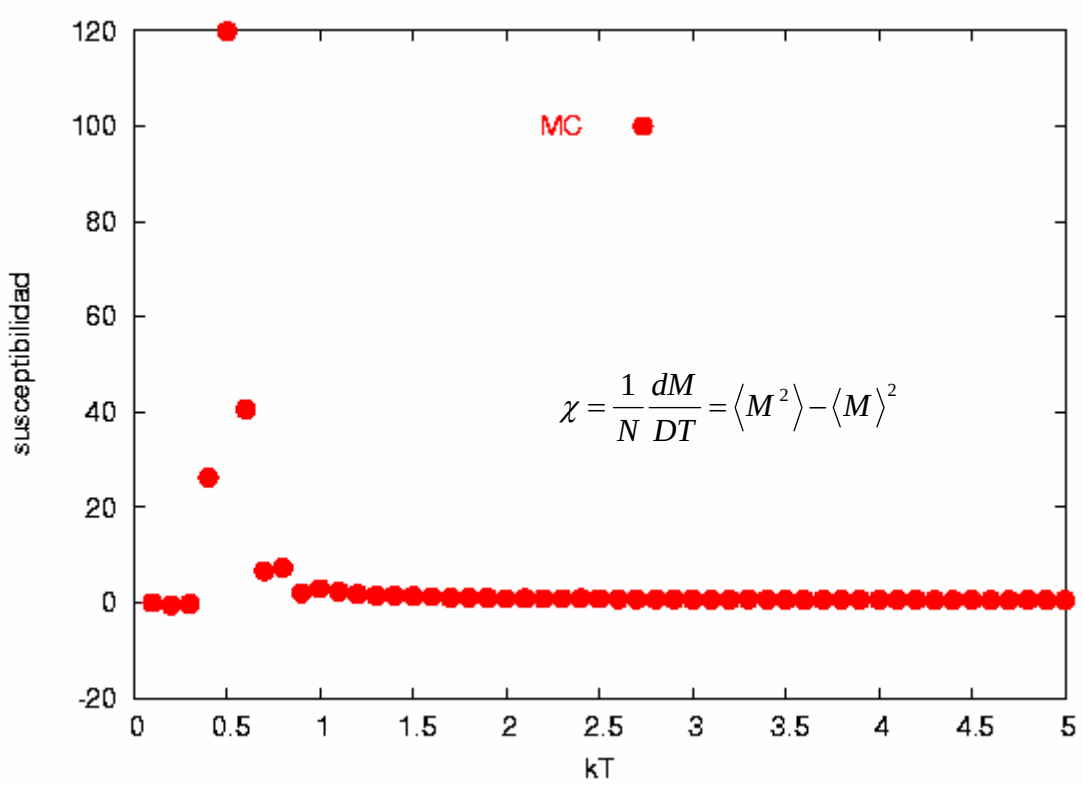


Fig. 5.4 The Monte Carlo susceptibility for the Ising case

For the case of Q>2, our results are displayed in figures 1.4 and 2.4 reference network (all edge lengths = 1) was compared with the network for treated cells (edge lengths = Euclidean distances off differential expressions between treated and reference cells). We collected the optimal temperatures for each network by determining the super-paramagnetic phases where the first granulations of the networks took place (Fig. 6.4 and 7.4)

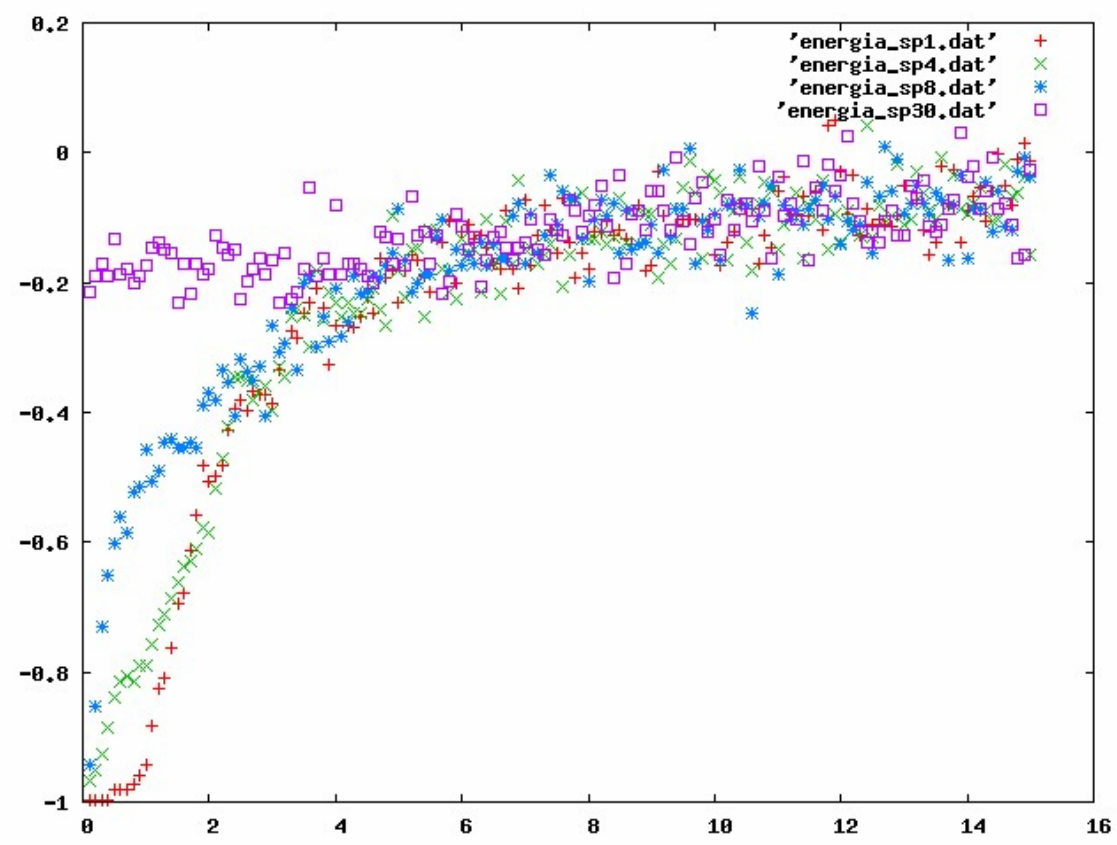


Fig 6.4. Energy for different temperatures.

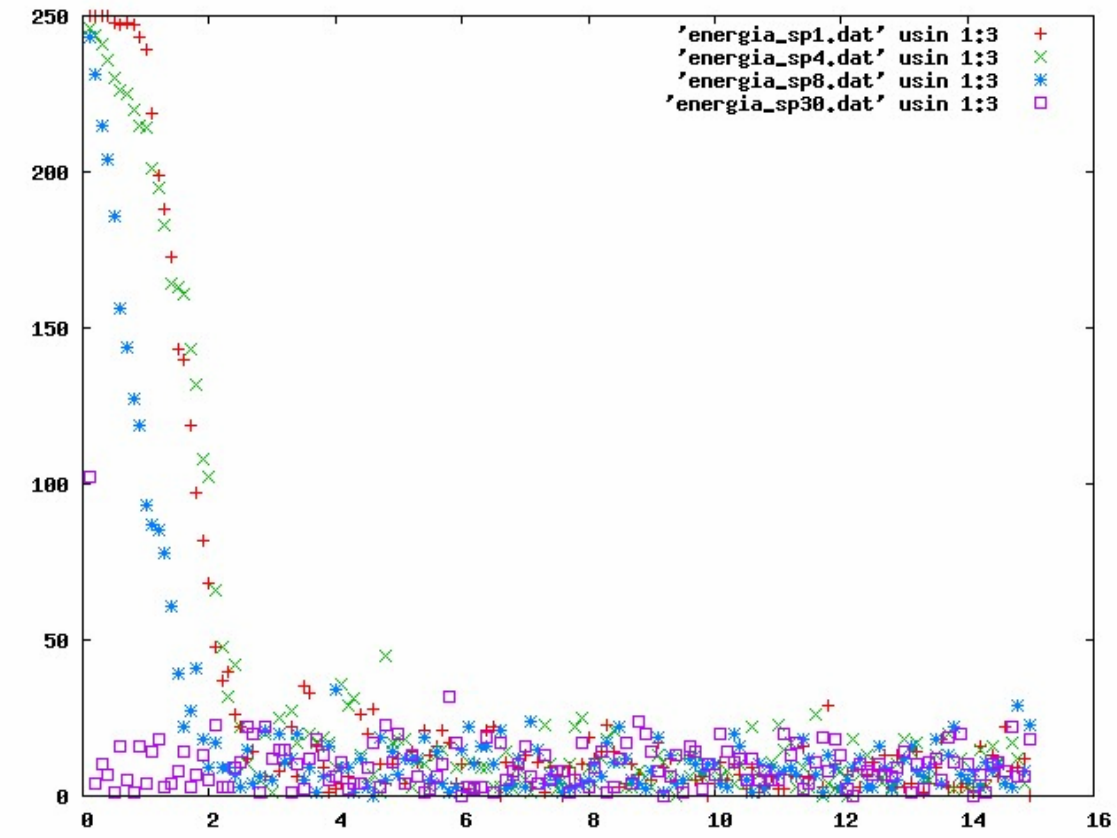


Fig. 7.4 Mean magnetization for different Qs.

References

[1] Stears, R.L.,Trends in Microarray analysis. *Nature Medicine*, **9**, 140-145 (2003).

[2] Botstein, D., Brown P. Exploring the new world of the genome with DNA microarrays. *Nature Genetics*, **21(suppl)**, 33-37 (1999).

[3] Stryer, L., *Biochemistry*, W. H. Freeman Company, New York (1995).

[4] Karp, P.D., Riley, M., Paley, S.M. and Pellegrini-Toole, A. The MetaCyc database. Nucleic Acids Res., **30**, 59-61 (2002).

[5] Zien, A., Küffner, R., Zimmer, R. and Lengauer, T. Analysis of gene expression data with pathway score. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **8**, 407-417 (2000).

[6] Eisen, M.B., Spellman P.T, Brown P.O., and Botstein D. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. US.A,* **95**, 14863-1486 (1998).

[7] Blatts, M., Wiseman, S. and Domany, E. Super-paramagnetic clustering of data. *Phys. Rev. Lett.* **76**, 3251- 3255 (1996).

[8] Brazman, A. and Vilo, J. Gene expression data analysis. *FEBS Lett.* **480**, 17-24 (2000).

[9] Himanshu A. and Domany E., Potts Ferromagnets on Coexpressed Gene Networks: Identifying Maximally Stable Partitions. Phys. Rev. Lett. 90, 158102 (2003).

[10] Dopazo, J., Zanders, E., Dragoni I., Amphlett, G. and Falciani, F. Methods and approaches in the analysis of gene expression data. *J. Immunol Methods* **250**, 93-112 (2001).

[11] Rose K., Gurewitz E. and Fox G.C. Statistical mechanics and phase transitions in clustering. Phys. Rev. Lett. 65, 945-948 (1990).

[12] Khodursky, A.B., Peter, B.J., Cozzarelli, N.R., Botstein, D., Brown, P.O. and Yanofsky, C. DNA microarray analysis of gene expression in response to physiological and genetic changes that affect tryprophan metabolism in Escherchia coli, *Proc. Natl. Acad. Sci. USA*, **97**, 12170-12175 (2000).

[13] Binder K. and Heermann D.W. Monte Carlo simulations in statistical physics: An introduction. Berlin: Springer-Verlag (1988).

[14] Wang S. and Swendsen R.H. Cluster Monte Carlo algorithms. Physica A **167**, 565-579 (1990).

[15] Swendsen R.H., Wang S. and Ferrenberg A.M. New Monte Carlo methods for improved efficiency of computer simulations in statistical mechanics. In K. Binder (Ed.), The Monte Carlo Method in Condensed Matter Physics (pp 75-91). Berlin: Springer-Verlag (1992).

[16] Gould H. and Tobochnik J. Overcoming critical slowing down. Computers in Physics, 29, 82-86 (1989).

[17] Billoire A., Lacaze R., Morel A., Gupta S., Irback A. and Petersson B. Dynamics near a first-order phase transition with the Metropolis and Swendsen Wang algorithms. *Nuclear Physics*, B358, 231-248 (1991).

[18] Hennecke M. and Heyken U. Critical-dynamics of cluster algorithms in the dilute Ising-model. *Journal of Statistical Physics*, 72, 829-844. (1993).

Chapter 5
High-gain nonlinear observer for simple genetic regulation

High-gain nonlinear observers occur in the nonlinear automatic control theory and are in standard usage in chemical engineering processes. We apply such a type of analysis in the context of a very one-gene regulation circuit. In general, an observer combines an analytical differential-equation-based model with partial measurement of the system in order to estimate the non-measured state variables. We use on the simplest observers, that of Gauthier et. al., which is a copy of the original system plus a correction term which is easy to calculate. For the illustration of this procedure, we employ a biological model, recently adapted from Goodwin's old book by De Jong, in which one plays with the dynamics of the concentrations of the messenger RNA coding for a given protein, the protein itself, and a single metabolite. Using the observer instead of the metabolite, it is possible to rebuild the non-measured concentrations of the mRNA and the protein.

5.1 Introduction

According to textbooks, gene expression is a very complicated dynamical process which is regulated at a number of its stages during the synthesis of proteins [1]. Similar to many big cities, with heavy traffic, biological cells host complicated traffic of biochemical signals at all levels. At the nanometer scale, clusters of molecules in the form of proteins drive the dynamics of the cellular network that schematically can be divided into four regulated parts: the DNA or genes, the transcribed RNAs, the set of interacting proteins, and the metabolites. Genes can only affect other genes through specific proteins, as

well as through some metabolic pathways that are regulated by proteins themselves. They act to catalyze the information stored in DNA, all the way from the fundamental processes of transcription and translation to the final quantities of produced proteins.

For the purpose of modeling, it is essential to generate simple models that help to understand elementary dynamical components of these complex regulatory networks as molecular tools that participate in an important way in the machinery of cellular decisions, that is to say, in the behavior and genetic program of cells. The central importance of control theory in biology can be assessed through the recent problem of identifying control motifs (or modules), which are patterns that occur in a gene network far more often than in randomized networks of biological regulators [2]. This hot issue has been first pinpointed in a breakthrough paper of Doyle et al. [3] in which the regulation of bacterial chemotaxis was interpreted in terms of the simple integral control "adaptive module" introduced by Barkai and Leibler [4]. Since gene regulation appears to occur only at some definite states of the whole process, which in general are not well known, we are from the point of view of control engineering in the case of the reconstruction of those specific states under the condition of limited information.

It is quite clear that the availability of all state variables to direct measurement is an extremely rare occasion for gene expression phenomena or when it is possible it could be too expensive. For this particular task, but in completely different technological areas, the engineers have developed software sensors (state observers) that accurately reconstruct the state variables of various technological processes [5]. The basic concept of state of a system or process could have many different empirical meanings in biology. For the particular case of gene expression, the meaning of a state is essentially that of a concentration. The typical problem in control engineering that appears to be tremendously useful in biology is reconstruction of some specific regulated states under conditions of limited information.

In general, an observer is expected to provide a good estimate $\hat{X}(t)$ of the natural state *X(t)* of the original system. For this, one usually can think that some distance $d(\hat{X}(t), X(t))$ (in the sense of a norm || . || in a vectorial space) goes to zero as $t \to \infty$. Such observers can be constructed using the mathematical model of the process to obtain an estimate $\hat{X}$ of the true state *X*. This estimate can then be used as a substitute for the unknown state *X*. The usage of state observers has proven useful in process monitoring and for many other tasks. The concept of observer is used herein in the sense of control theory, defining an algorithm capable of giving a reasonable estimation of the unmeasured variables of a process. In the case of gene expression processes the description in made very concrete in the following by looking at quite simple mathematical models that refer to single gene cases and which in principle can be extended to some *operons* that are single gene clusters.

In this chapter, we will examine in detail a particularly simple observer due to Gauthier et al. [6] possessing arbitrary exponential decay and linear error

dynamics for the case of a three-state genetic regulation process. We were led to consider this observer because of its simplicity and its *high-gain* property. The gain is defined as the amount of increase in error in the dynamics of the observer. This amount is directly related to the velocity with which the observer recovers the unknown signal. For the observer of Gauthier et al. the amount of increase in error is constant and usually of high values leading to a fast recover of the unmeasurable states.

5.2 Mathematical model for a simple gene regulation process

A kinetic model of a simple genetic regulation process was first developed by Goodwin as long as 1963 [7]. It has been further generalized by Tyson and Othmer [8] and clearly explained by De Jong in his recent review [9]. We consider here the simplest version of this kinetic model. For three concentrations X_1, X_2, X_3, corresponding to the messenger RNA (mRNA) that codes for the unstable enzyme, the enzyme, and the metabolite, respectively, we write Tyson's model in the form

$$\Gamma_{biology}\begin{cases}\dot{X}_1 = K_1 H(x,\vartheta) - \gamma_1 X_1, \\ \dot{X}_2 = K_2 X_1 - \gamma_2 X_2, \\ \dot{X}_3 = K_3 X_2 - -\gamma_3 X_{3.}\end{cases} \tag{1.5}$$

The parameter K_1, K_2, K_3 are all strictly positive and represent production constants, whereas $\gamma_1, \gamma_2, \gamma_3$ are also strictly positive degradation constants. These rate equations express a balance between the number of molecules appearing and disappearing per unit time. Notice that the model assumes that the concentration X_2 increases linearly with X_1 and the concentration X_3 linearly X_2, which are natural assumptions. In the case of X_1, the first term is the production term involving a nonlinear non-dissipative *regulation function H* that we take of the *m*-steepen Hill form *(m>0* is the steepness parameter) in common use:

$$H^+(X,\vartheta) = \frac{X_3^{\,m}}{X_3^{\,m} + \vartheta^m}, \qquad H^-(X,\vartheta) = 1 - \frac{X_3^{\,m}}{X_3^{\,m} + \vartheta^m} \tag{2.5}$$

for the activation and inhibition cases, respectively. The parameter θ gives the threshold for the regulatory influence of the concentration of the metabolite on the target gene, whereas the steepness parameter *m* is a measure of the collective effect of groups of metabolite molecules and also defines the shape of the Hill curve. This nonlinear parametrization describes the "biological regulation process" that includes the production of the mRNA by transcription of its structural gene, its possible intranuclear processing by cleavage, its enzymatic degradation within the nucleus, and its migration to the cytoplasm by

some form of diffusion or biological transport. Once in the cytoplasm, the mRNA is both translated into the unstable enzyme and enzymatically degraded.

System $\Gamma_{bio\log y}$ and its trivial chain generalization in the linear part is considered to be a good model for the simplest type of allosteric regulation in biochemistry, i.e., the inhibition or activation of an enzyme or protein by a small regulatory molecule that interacts with the enzyme at a site (allosteric site) other than the active site at which catalytic activity occurs. The interaction changes the shape of the enzyme, thus affecting the active site of the standard catalysis. This change of shape of the enzyme is sufficient to change its ability to catalyze a reaction in either negative or positive way and enables a cell to regulate needed metabolites. The allosteric regulation has the typical features of a feedback loop in control theory if the regulatory protein acts on the enzyme in the pathway of its own synthesis.

5.3 The nonlinear observer

Many attempts have been made to develop nonlinear observer design methods. One could mention the industrially popular extended Kalman filter, whose design is based on a local linearization of the system around a reference trajectory, restricting the validity of the approach within a small region in the state space [5-10]. The first systematic approach for the development of a theory of nonlinear observers was proposed same time ago by Krener and isidori [11]. In further works, nonlinear transformations of the coordinates have also been employed to put the considered nonlinear system in a suitable "observer canonical form", in which the observer design problem may be easily solved [6, 12, 13]. The main idea in this case is to find a state transformation to represent the system as a linear differential equation plus a nonlinear term, is a function of the measured state.

In this section, we present the design of a nonlinear software sensor in which the metabolite concentration is the naturally measured state (the most easy to measure) and corresponds to the mathematical state X_3 in the model introduced in the previous section. Therefore, it seems logical to take X_3 as the output of the system

$$Y=h(X) =X_3. \tag{3.5}$$

We now apply the technique of high-gain observes that works for many nonlinear systems and guarantees that the output feedback controller recovers the performance of the state feedback controller when the observer's gain is sufficiently high. The model given by the aforementioned system $\Gamma_{bio\log y}$ has the form

$$\Gamma_y : \begin{cases} \dot{X} = f(X), \\ y = h(X) \end{cases} \tag{4.5}$$

In which $X \in \mathrm{R}^3$, and moreover there is a "physical subset" $\Omega \subset \mathrm{R}^3$ where the system lies. To make this mathematically precise we must introduce some further mathematical terminology. Let us construct the *j*th time derivative of the output. This can be expressed using Lie differentiation of the function *h* by the vector field *f*, $L_f^j(h)(X(t))$. $L_f^j(h)(X(t))$ is the *j*th Lie derivative of *h* by *f* and a function of *X* defined inductively as follows:

$$\begin{aligned} &L_f^0(h)(X) = h(X), \\ &L_f^j(h)(X) = \frac{\partial}{\partial X}(L_f^{j-1}(h)(X))\, f(X). \end{aligned} \tag{5.5}$$

When Γ_y is observable, the map $\Phi : X \to \Phi(X)$ is a diffeomorphism where

$$\xi = \Phi(X) = \begin{pmatrix} L_f^0(h)(X) \\ L_f^1(h)(X) \\ L_f^2(h)(X) \end{pmatrix} = \begin{bmatrix} X_3 \\ K_3 X_2 - \gamma_3 X_3 \\ K_3(K_2 X_1 - \gamma_2 X_2) - \gamma_3(K_3 X_2 - \gamma_3 X_3 \end{bmatrix}. \tag{6.5}$$

For $\Phi(X)$ to be a local diffeomorphism in a region Ω, it is necessary and sufficient that the Jacobian $\mathrm{d}\Phi(X)$ should be non-singular on Ω and moreover that $\Phi(X)$ is one-to-one from Ω to $\Phi(X)$, see [14]. Notice that no matter if we choose $H^+(X, \vartheta)$, the coordinate transformation is the same. This means that the structure of the observer will be the same for both cases: gene activation or inhibition.

When the system is observable on Ω, it can be rewritten in the global coordinate system defined by $\Phi(X)$ in the following matrix form:

$$\begin{aligned} &\Gamma_y' : \begin{cases} \dot{\xi} = F'(\xi) = \begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \end{bmatrix} = \begin{bmatrix} \xi_2 \\ \xi_3 \\ \varphi(\xi) \end{bmatrix}, \\ y = C\xi = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix}, \end{cases} \end{aligned} \tag{7.5}$$

where, moreover, φ can be extended from Ω to the entire R^3 by a C^∞ function globally Lipschitz on R^3. The latter form allows us to make use of the following result proven by Gauthier et al. [6]:

Consider the system

$$\Gamma_G : \dot{\hat{\xi}} = F'(\hat{\xi}) - S^{-1}C^T(C\hat{\xi} - y), \tag{8.5}$$

where $S(\theta)$ is the solution of the matrix equation

$$\theta S - A^T S - SA + C^T C = 0 \tag{9.5}$$

for θ large enough, with A a matrix of Brunovsky form ($A=\delta_{ij+1}; \delta_{ij}$ is the Kronecker symbol), which plays the role of a shift operator on R^3. Then Eq. (8.5) defines an observer for Γ'_y with

$$\left\|\hat{\xi} - \xi\right\| \leq M \exp\left(-\frac{0}{3}t\right)\left\|\hat{\xi}_0 - \xi_0\right\|. \tag{10.5}$$

In our case, an observer is a dynamical system as given by Eq. (8.5) that Hill tracks the trajectory of the original system (here Γ'_y). Notice that both systems are identical unless for an additional term that compensates the error in the observer, where the error is given by the difference $\left\|\hat{\xi} - \xi\right\|$, which is seen to be exponentially decreasing in time. The Gauthier observer is particularly simple since it appears to be only a copy of Γ'_y, together with a correction term that depends only on the dimension of the state space and not on the system Γ'_y itself. In others words, the structure of the observer does not depend on the Hill steepness parameter *m* (Eq. (1.5)).

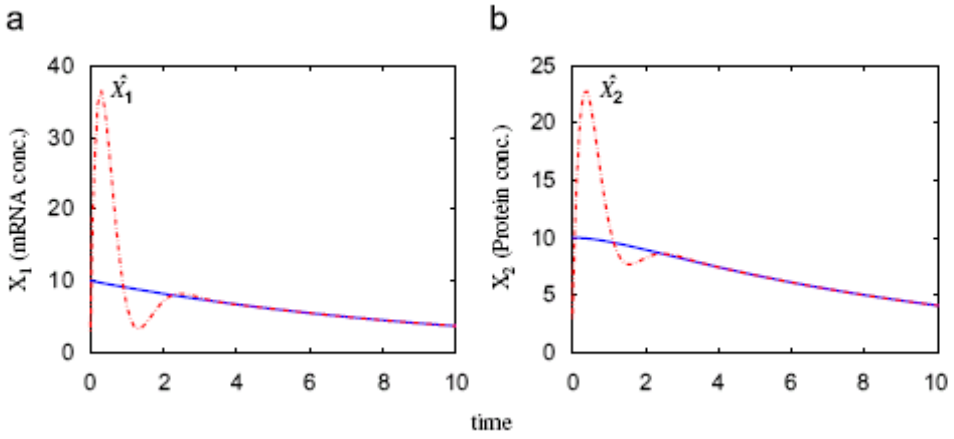


Figure 1.5 The numerical simulation-solid lines represent the true states and dotted lines represent the Gauthier estimates given by Eq. (15) for an activation case. Plots (a) represents the evolution of mRNA concentration in time and plot (b) the variation of protein (enzyme) concentration in time.

For the sake of concreteness we will construct the observer only for the activation case. However, one should notice that only the function $f(\hat{X})$ will change for the inhibition case.

The Gauthier observer in Eq. (8.5) in the original coordinates is given by

$$\dot{\hat{X}} = f(\hat{X}) + \Upsilon(\hat{X})S^{-1}C^{T}(h(X) - h(\hat{X})), \tag{11.5}$$

where

$$\Upsilon(\hat{X}) = \left.\frac{\partial \Phi^{-1}}{\partial \hat{\xi}}\right|_{\hat{\xi}=\Phi(\hat{\xi})}. \tag{12.5}$$

For the particular three-dimensional state space of $\Gamma_{bio\log y}$ we get

$$\Upsilon(\hat{X}) = \begin{bmatrix} \frac{\gamma_2\gamma_3}{K_2K_3} & \frac{\gamma_2+\gamma_3}{K_2K_3} & \frac{1}{K_2K_3} \\ \frac{\gamma_3}{K_3} & \frac{1}{K_3} & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{13.5}$$

The matrix $S(\theta)$ in the three-dimensional case can be computed by means of Eq. (9.5) given in the Gauthier theorem and its inverse $S^{-1}(\theta)$ appears to be

$$S^{-1}(\theta) = \begin{bmatrix} 3\theta & 3\theta^2 & \theta^3 \\ 3\theta^2 & 5\theta^3 & 2\theta^4 \\ \theta^3 & 2\theta^4 & \theta^5 \end{bmatrix}. \tag{14.5}$$

Plugging matrices (13) and (14) in Eq. (11), we get the following equation for the observer introduced by Gauthier et al. applied to our biological case:

$$\dot{\hat{X}}_{bio\log y} = f(\hat{X}) + \left[3\frac{\gamma_2\gamma_3\theta}{K_2K_3} + 3\frac{(\gamma_2+\gamma_3)\theta^2}{K_2K_3} + \frac{\theta^3}{K_2K_3}\right](X_3 - \hat{X}_3). \tag{15.5}$$

We use this form of the Gauthier observer to estimate the states X_1 and X_2 of the dynamical system $\Gamma_{bio\log y}$. We work with $\theta=1$ and the values of the parameters given in Table 1 that are not necessarily the experimental values but are consistent with the requirements of the model. Fig. **1.5** shows the results of a numerical simulation, where the solid lines represent the true states and the dotted lines stand for the estimates, respectively. In addition, for the real system we have taken $m=2$ whereas for the observer $m=1$ in order to show the

robustness of this type of nonlinear observer with respect to the steepness parameter.

Symbol	Meaning	Value (arb. units)
K_1	Production constant of mRNA	0.001
K_2	Production constant of protein A	1.0
K_3	Production constant of metabolite K	1.0
γ_1	Degradation constant of mRNA	0.1
γ_2	Degradation constant of protein A	1.0
γ_3	Degradation constant of metabolite K	1.0
ϑ	Hill´s threshold parameter	1.0
m	Steepness parameter	2.0

Table 1.5 Parameters of the Goodwin biological model used in this chapter.

5.4 Conclusion

We represented here the mathematical exercise of designing a high-gain observer for a simple one-gene regulation dynamic process involving end-product activation (inhibition leads to similar results), which is able to rebuild in an effective way the non-measured concentrations of mRNA and the involved protein. Thus, the limitation of those experiments in which one has only the metabolite available can be overcome by employing this simple observer. In addition, this type of nonlinear observer could be used on line and is robust with respect to m, i.e., it does not need the exact value of the Hill steepness parameter. However, for more complex input of more complicated observable dynamical systems, this constant gain observer could have less performance and be overcome by some *adaptive* observers that can change in order to work better or provide more fit for a particular purpose. In the case of more limited information, e.g., for unknown functional form of the regulation function and high noise levels that can spoil the performance of the observer, the completely different mathematical procedure of creating dynamical extension of the observer system is required [15].

References

[1] B. Lewin, Genes VII, Oxford University Press, Oxford, 1999.

[2] S. Mangan, U. Alon, Proc. Natl. Acad. Sci. USA 100 (2003) 11980-11985.

[3] T.M. Yi, Y. Huang, M.I. Simon, J. Doyle, Proc. Natl. Acad. Sci. USA 97 (2000) 4649-4653.

[4] N. Barkai, S. Leibler, Nature 387 (1997) 913-917.

[5] G. Stephanopoulos, Chemical Process Control, Prentice-Hall, Englewood Cliffs, NJ, 1984.

[6] J.P. Gauthier, H. Hammouri, S. Othman, IEEE Trans. Autom. Control 37 (1992) 875-880.

[7] B.C. Goodwin, Temporal Oscillations in Cells, Academic Press, New York, 1963.

[8] J.J. Tyson, H.G. Othmer, Prog. Theor. Biol. 5 (1978) 1-62.

[9] H. De Jong, J. Comput. Biol. 9 (2002) 67-103.

[10] W.A. Wolovich, Automatic Control System, Saunders College Publishing, Fort Worth, 1994.

[11] A.J. Krener, A. Isidori, Syst. Control Lett. 3 (1983) 47-52

[12] J.P. Gauthier, G. Bornard, IEEE Trans. Autom. Control AC-26 (1981) 922-926.

[13] J.P. Gauthier, I.A.K. Kupta, SIAM J. Control Optimization 32 (1994) 975-994.

Chapter 6
Nonlinear Software Sensor for Monitoring Genetic Regulation

Nonlinear control techniques by means of a software sensor that are commonly used in chemical engineering could be also applied to genetic regulation processes. We provide here a realistic formulation of this procedure by introducing an additive white Gaussian noise, which is usually found in experimental data. Besides, we include model errors, meaning that we assume we do not know the nonlinear regulation function of the process.

In order to illustrate this procedure, we employ the Goodwin dynamics of the concentrations [B. C. Goodwin, Temporal Oscillations in Cells (Academic, New York, 1963)] in the simple form recently applied to single gene systems and some operon cases [H. De Jong, J. Comput. Biol. 9, 67 (2002)] which involves the dynamics of the mRNA, given protein, and metabolite concentrations. Further, we present results for a three gene case in coregulated sets of transcription units as they occur in prokaryotes. However, instead of considering their full dynamics, we use only the data of the metabolites and a designed software sensor. We also show, more generally, that it is possible to rebuild the complete set of nonmeasured concentrations despite the uncertainties in the regulation function or, even more, in the case of not knowing the mRNA dynamics.

In addition, the rebuilding of concentrations is not affected by the perturbation due to the additive white Gaussian noise and also we managed to filter the noisy output of the biological system.

6.1 Introduction

Gene expression is a complex dynamic process with intricate regulation networks all along its stages leading to the synthesis of proteins [1]. At the present time, its best studied regulation feature is the DNA transcription. Nevertheless, the expression of a gene should be also regulated during the RNA processing and transport, RNA translation, and also in the posttranslational modification of proteins. Control engineering is a key discipline with tremendous potential to simulate and manipulate the processes of gene expression. In general, the control terminology and its mathematical methods are poorly known to the majority of biologists. Many times the control ideas are simply reduced to the homeostasis concept. However, the recent launching of the IEE journal Systems Biology [2] points to many promising developments from the standpoint of systems analysis and control theory in biological sciences. Papers like that of Yi *et. al* [3], in which the Barkai and Leibler robustness model [4] of perfect adaptation in bacterial chemotaxis is shown to have the property of a simple linear integral feedback control, could be considered as pioneering work in the field.

We mention here two important issues. The first one is that the basic concept of state of a system or process could have many different empirical meanings in biology. For the particular case of gene expression, the meaning of a state is essentially that of a concentration. The typical problem in control engineering that appears to be tremendously useful in biology is the reconstruction of some specific regulated states under conditions of limited information. Moreover, equally interesting is the issue of noise filtering. It is quite well known that gene expression is a phenomenon with two sources of noise: one due to the inherent stochastic nature of the process itself and the other originating in the perturbation of the natural signal due to the measuring device. In the mathematical approach, the latter class of noise is considered as an additive contamination of the real signal and this is also our choice here. Both issues will form the subject of this investigation.

Taking into account the fact that rarely one can have a sensor on every state variable, and some form of reconstruction from the available measured output data is needed, software can be constructed using the mathematical model of the process to obtain an estimate $\hat{X}$ of the true state X . This estimate can then be used as a substitute for the unknown state X . Ever since the original work by Luenberger [5], the use of state observers has proven useful in process monitoring and for many other tasks. We will call herein as observer, in the sense of control theory, an algorithm capable of giving a reasonable estimation of the unmeasured variables of a process. For this reason, it is widely used in control, estimation, and other engineering applications. Since almost all observer designs are heavily based on mathematical models, the main drawback is precisely the dependence of the accuracy of such models to describe the naturally occurring processes. Details such as model uncertainties and noise could affect the performance of the observers. Taking into account these details is always an important matter and should be treated carefully. Thus, we will pay special attention in this research to estimating unknown states of the gene expression process under the worst possible case, which

corresponds to noisy data, modeling errors, and unknown initial conditions. These issues are of considerable interest and our approach is a novel contribution to this important biological research area. Various aspects of noisy gene regulation processes have been dealt with recently from both computational and experimental points of view in a number of interesting papers [6]. We point out that since we add the noise δ to the output of the dynamic system in the form $y = CX + \delta$ (see Eqs. Γ in Section 6.4) it seems that its origin is mainly extrinsic to the regulation process, even though it could be considered as a type of intrinsic noise with respect to the way the experiment is performed. On the other hand, when writing the equation in the form $y = C(X + I\Delta)$, where Δ is a vector of noisy signals, one can see that the observer could estimate states that are intrinsically noisy even though the processes are still deterministic.

6.2 Brief on the biological context

Similar to many big cities, with heavy traffic, biological cells host complicated traffic of biochemical signals at all levels. Like cars on a busy highway, millions of molecules get involved in the bulk of the cell in many life processes controlled by genes. At the nanometer level, clusters of molecules in the form of proteins drive the dynamics of the cellular network that schematically can be divided into three self regulated parts: the genes, the set of interacting proteins and the metabolites. Genes can only affect other genes through special proteins, as well as through some metabolic pathways that are regulated by proteins themselves. They act to catalyze the information stored in DNA, all the way from the fundamental processes of transcription and translation to the final quantities of produced proteins.

Considering the enormous complexity of multicellular organisms generated by their large genomes, one can nevertheless still associate at least one regulatory element to any component gene. Each regulatory element integrates the activity of at least two other genes. This is how the functioning of complex regulatory transcriptional and translational networks is understood at the present time [7, 8]. However, one can hope that this extraordinary complexity can be summarized, at least at some levels, by simplified models which can help get insight in the inner processes of the biological networks.

Many entities in cellular networks can be identified as the basic units of regulation, mainly distinguished by their unique roles with respect to interaction with other units. These basic units are the genes, the proteins that the genes can produce, the forms of each protein, protein complexes, and all related metabolites. These units have associated values that either represents concentrations or levels of activation. These values depend on both the values of the units that affect them due to the aforementioned mechanisms and on some parameters that govern each special form of interaction.

This gives rise to genetic regulatory systems structured by networks of regulatory interactions between DNA, RNA, proteins, and small molecules. The simplest regulatory network is made of only one gene going into its

transcriptional process and then passing to the translation of its mRNA into proteins, and further to the catalytic stage. This is when appropriate enzymes turn specific metabolites into those ones that are capable to activate repressor proteins towards their final action onto the gene itself. This simple regulatory system is actually what is called a feedback loop in control engineering. A mathematical model of such a biological inhibitory loop has been discussed since a long time ago by Goodwin and recurrently occurred in the literature, most recently being reformulated by De Jong [9]. Although this case could look unrealistic, there are simple organisms, such as bacteria, where one regulatory loop may prove essential as recently discussed in detail by Ozbudak et al [10]. However, already at the level of two genes the situation gets really complicated, mostly because of the possible formation of heterodimers between the repressors and other proteins around. These heterodimers are able to bind at the regulatory sites of the gene and therefore can affect it and lead to modifications of the regulatory process. Recent developments of experimental techniques, like cDNA microarrays and oligonucleotide chips, have allowed rapid measurements of the spatiotemporal expression levels of genes [11-13]. In addition, formal methods for the modeling and simulation of gene regulation processes are currently being developed in parallel to these experimental tools. As most genetic regulatory systems of interest involve many genes connected through interlocking positive and negative feedback loops, an intuitive understanding of their dynamics is hard to obtain. The advantage of the formal methods is that the structure of regulatory systems can be described unambiguously, while predictions of their behavior can be made in a systematic way.

To make the description very concrete, it is interesting to look at well-defined, i.e., quite simple mathematical models that we present in the next section that refers to single gene cases and single gene clusters (operons). The nonlinear software sensor for such cases is discussed in Section IV. A three-gene case is treated as an extension to regulatory gene networks and shows that the method of forward engineering still works for reasonably simple gene networks. The conclusion section comes at the end of the paper.

6.3 Mathematical model for gene regulation

In this section, we use the very first kinetic model of a genetic regulation process developed by Goodwin in 1963 [15], generalized by Tyson in 1978 [16] and most recently explained by De Jong [9]. The model in its most general form is given by the following set of equations:

$$\dot{X}_1 = K_{1n} r(X_n) - \gamma_1 X_1, \tag{1.6}$$

$$\dot{X}_i = K_{i,i-1} X_{i-1} - \gamma_i X_i, \quad 1 < i \leq n. \tag{2.6}$$

The parameters $K_{1n}, K_{21}, ..., K_{n,n-1}$ are all strictly positive and represent production constants, whereas $\gamma_1, ..., \gamma_n$ are strictly positive degradation constants. These rate equations express a balance between the number of molecules appearing and disappearing per unit time. In the case of X1, the first

term is the production term involving a nonlinear nondissipative regulation function. We take this as an unknown function. On the other hand, the concentration X_i, $1 < i \leq n$ increases linearly with X_{i-1}. As well known, in order to express the fact that the metabolic product is a co-repressor of the gene, the regulation function should be a decreasing function for which most of the authors use the Hill sigmoid, the Heaviside and the logoid curves. The decrease of the concentrations through degradation, diffusion and growth dilution is taken proportional to the concentrations themselves. For further details of this regulation model we recommend the reader the review of De Jong [9]. It is to be mentioned here that bacteria have a simple mechanism for coordinating the regulation of genes that encode products involved in a set of related processes: these genes are clustered on the chromosome and are transcribed together. Most prokaryotic mRNAs are polycistronic (multiple genes on a single transcript) and the single promoter that initiates transcription of clusters is the site of regulation for expression of all genes in the cluster. The gene cluster and promoter, plus additional sequences that function together in regulation, are called operon. Operons that include two to six genes transcribed as a unit are common in nature [17].

The fact that two or more genes are transcribed together on one polycistronic mRNA implies that we have a unique mRNA production constant and consequently we also have one mRNA degradation constant. In addition, the polycistronic mRNA can translate an enzyme, resulting in the existence of just one enzyme production and degradation constant, respectively. The same applies for the metabolite produced through the enzyme catalysis. Thus, if the resulting metabolite has repressor activity over the polycistronic mRNA (as in the case of tryptophan [18]), then the model given by Eqs. (1) and (2) could also be applied to operons and therefore it has a plausible application to the study of prokaryotic gene regulation.

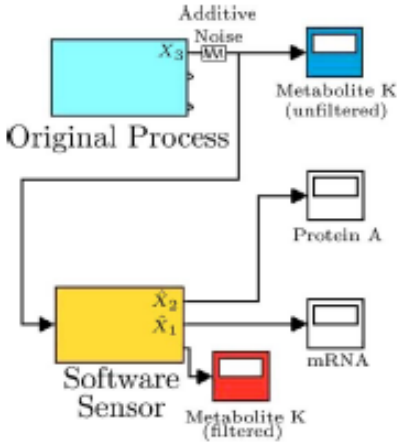


Figure 1.6 Schematic representation of the software sensor, where the output of the system is the input of the software sensor and the outputs of the latter are the rebuilt concentrations.

6.4 The nonlinear Aguilar observer

Numerous attempts have been made to develop nonlinear observer design methods. One could mention the industrially popular extended Kalman filter, whose design is based on a local linearization of the system around a reference trajectory, restricting the validity of the approach to a small region in the state space [14, 19]. The first systematic approach for the development of a theory of nonlinear observers was proposed some time ago by Krener and Isidori [20]. In further research, nonlinear transformations of the coordinates have also been employed to put the considered nonlinear system in a suitable observer canonical form", in which the observer design problem may be easily solved [21-23]. Nevertheless, it is well known that classical proportional observers tend to amplify the noise of on-line measurements, which can lead to the degradation of the observer performance.
In order to avoid this drawback, this observer algorithm is based on the works of Aguilar et al. [24-25], because the proposed integral observer provides robustness against noisy measurement and uncertainties. We show that this new structure retains all the characteristics of the popular (the traditional high gain) state observers of the classical literature and furthermore provides additional robustness and noise filtering and thus can result in a significant improvement of the monitoring performances of the genetic regulation process.

In this section, we present the design of a nonlinear software sensor in which one X_j, for j ϵ (1… *n*), is the naturally measured state (the easiest to measure). Therefore, it seems logical to take X_j as the output of the system

$$y = h(X) = X_j \tag{3.6}$$

Now, considering the constant K_{1n} and the function $r(X_n)$ as unknown, we group them together in a function $\tau(X)$. In addition, we consider that the output function $h(X)$ is contaminated with a Gaussian noise. In such a case, the model given by the aforementioned, Eqs. (1) and (2) acquires the form

$$\Gamma : \begin{cases} \dot{X} = \bar{\tau}(X) + l(X), \\ y = CX + \delta, \end{cases}$$

where $\bar{\tau}$ is a n X 1 vector whose first entry is $\tau(X)$ and all the rest are zero, $l(X)$ is also a n X 1 vector of the form $\left[-\gamma_1 X_1, K_{i,i-1} X_{i-1} - \gamma_i X_i\right]^T, \delta,$ is an additive bounded measurement noise, and $X \in R^n$.The system is assumed to lie in a "physical subset" $\Sigma \subset R^n$.

Then, the task of designing an observer for the system Γ is to estimate the vector of states X, despite of the unknown part of the nonlinear vector $\bar{\tau}(X)$ (which should be also estimated) and considering that y is measured on-line and that the system is observable.

A particular representation of the software sensor that we describe here is provided in Fig. **1.6**

In order to provide the observer with robust properties against disturbances, Aguilar and collaborators [24] considered only an integral type contribution of the measured error. Moreover, an uncertainty estimator is introduced in the methodology of observation with the purpose of estimating the unknown components of the nonlinear vector $\bar{\tau}(X)$. As a result, the following representation of the system is proposed:

$$\Xi : \begin{cases} \dot{X}_0 = CX + \delta, \\ \dot{X} = \bar{\tau} + l(X), \\ \dot{\bar{\tau}} = \Theta(X), \\ y_0 = X_0, \end{cases}$$

that is, in the case of the model given by Eqs. (1.6) and (2.6),

$$\begin{aligned} &\dot{X}_0 = X_j + \delta \\ &\dot{X}_1 = X_{n+1} - \gamma_1 X_1 \\ &\dot{X}_i = K_{i,i} X_{i-1} - \gamma_i X_i, \ 1 < i \leq n \\ &\dot{X}_{n+1} = \Omega(X) \\ &y = X_0, \end{aligned} \tag{4.6}$$

where $\dot{X}_0$ is the dynamical extension that allows us to integrate the noisy signal in order to recover a filtered signal, while $\dot{X}_{n+1}$ allows us to put the unknown regulation function as a new state. Thus, the task becomes the estimation of this new state (a standard task for an observer), and therefore the function Ω is related to the unknown dynamics of the new state. At this point, $X \in R^{n+2}$ and furthermore the following equation is generated

$$\dot{X} = AX + B + E\delta$$

where AX is the linear part of the previous system such that A is a matrix equivalent in form to a Brunovsky matrix, $B = [0,...,0,\Omega(X)]^T$ and $E = [1,0,...,0]^T$

We will need now the following result proven in Ref [24].

An asymptotic-type observer of the system Ξ is given as follows:

$$\hat{\Xi}:\begin{cases}\dot{\hat{X}}_0 = C\hat{X} + \theta_1(y_0 - \hat{y}_0),\\ \dot{\hat{X}} = \hat{\bar{\tau}} + l(\hat{X}) + \theta_2(y_0 - \hat{y}_0),\\ \dot{\hat{\bar{\tau}}} = \theta_3(y_0 - \hat{y}_0),\\ \hat{y}_0 = \hat{X}_0,\end{cases}$$

where the gain vector θ of the observer is given by

$$\theta = S_\theta^{-1} C^T,$$

$$S_{\theta;i,j} = \left(\frac{S_{i,j}}{\theta^{i+j+1}}\right)$$

Each entry of the matrix S_θ is given by the above equation, where S_θ is an n X n matrix (i and j run from 1 to n), and $S_{i,j}$ are entries of a symmetric positive definite matrix that do not depend on θ. Thus, $S_{i,j}$ are such that S_θ is a positive solution of the algebraic Riccati equation,

$$S_\theta\left(A + \frac{\theta}{2}I\right) + \left(A + \frac{\theta}{2}I\right)S_\theta = C^T C \tag{5.6}$$

In all formulas, $C = [1,0,...,0]$. In the multivariable case we must create one matrix S_θ for each block corresponding to each output. It is worth mentioning that we can think about this observer as a `slave' system that follows the `master' system, which is precisely the real experimental system. In addition, S_θ, as functional components of the gain vector, guarantees the accurate estimation of the observer through the convergence to zero of the error dynamics, i.e., the dynamics of the difference between the measured state and its corresponding estimated state. One can see that θ generates an extra degree of freedom that can be tuned by the user such that the performance of the software sensor becomes satisfactory for him.

In [28] it has been shown that such an observer has an exponential-type decay for any initial conditions. Notice that a dynamic extension is generated by considering the measured output of the original system as new additional dynamics with the aim to filter the noise. This procedure eliminates most of the noise in the new output of the system. The reason of the filtering effect is that the dynamic extension acts at the level of the observer as an integration of the output of the original system, (see the first equation of the system Ξ and the error part in the equations of system $\hat{\Xi}$). The integration has averaging effects upon the noisy measured states. More exactly, the difference between the integral of the output of the slave part of system $\hat{\Xi}$ and the integral of the output

of the original system gives the error and the observer is planned in such a way that the error dynamics goes asymptotically to zero, which results in the recovering of both the filtered state and the unmeasured states.

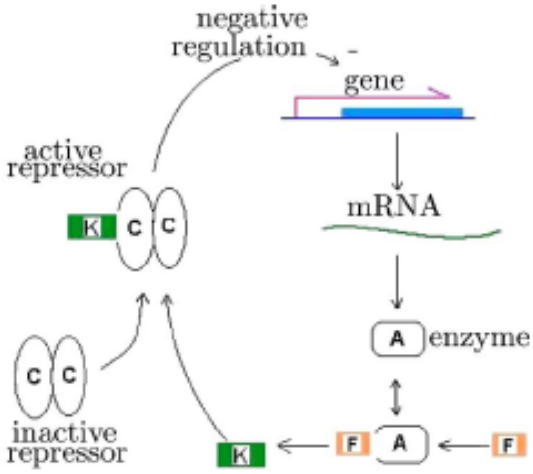


Figure 2.6 The genetic regulatory system given by Eqs. (6.6) - (6.8) involving end-product inhibition according to De Jong [9]. A is an enzyme and C a repressor protein, while K and F are metabolites. The mathematical model, as used by De Jong and by us, takes into account experiments where only metabolite K is measured.

A. Particular Case

For gene regulation processes, which are of interest to us here, we merely apply the aforewritten system of equations corresponding to the asymptotic observer $\hat{\Xi}$,

$$\dot{X}_1 = K_{1,3} r(X_3) - \gamma_1 X_1, \tag{6.6}$$

$$\dot{X}_2 = K_{2,1} X_1 - \gamma_2 X_2, \tag{7.6}$$

$$\dot{X}_3 = K_{3,2} X_2 - \gamma_3 X_3. \tag{8.6}$$

The pictorial representation of this system of equation is given in Fig. **2.6**

The values of the parameters given in Table **1.6**, without necessarily being the experimental values, are however consistent with the requirements of the model. Using the structure given by the equations of $\hat{\Xi}$, the explicit form of the software sensor is:

$$\dot{\hat{X}}_0 = \hat{X}_3 + \theta_1 (y_0 - \hat{X}_3),$$

$$\dot{\hat{X}}_1 = X_4 - \gamma_1 X_1 + \theta_2 (y_0 - \hat{y}_0),$$

$$\dot{\hat{X}}_2 = K_{2,1} X_1 - \gamma_2 X_2 + \theta_3 (y_0 - \hat{y}_0),$$

$$\dot{\hat{X}}_3 = K_{3,2} X_2 - \gamma_3 X_3 + \theta_4 (y_0 - \hat{y}_0),$$

$$\dot{\hat{X}}_4 = \theta_5 (y_0 - \hat{X}_3),$$

$$\hat{y}_0 = \hat{X}_0$$

Notice that this dynamic structure does not involve the regulation function.
We can solve Eq. (5.6) and for numerical purposes we choose ϑ = 2.5 and the standard deviation of the Gaussian noise of 0.001. Figure 3.6 shows the numerical simulation that illustrates the filtering effect of the software sensor over the noisy measured state.

Symbol	Meaning	Value (arb. units)
$K_{1,3}$	Production constant of mRNA	0.001
$K_{2,1}$	Production constant of protein A	1.0
$K_{3,2}$	Production constant of metabolite K	1.0
γ_1	Degradation constant of mRNA	0.1
γ_2	Degradation constant of protein A	1.0
γ_3	Degradation constant of metabolite K	1.0
ϑ	Hill´s threshold parameter	1.0

Table 1.6 Parameters of the Goodwin biological `signals' used in this chapter.

On the other hand, Fig. **4.6** shows the results of a numerical simulation, where the solid lines stand for the true states and the dotted lines indicate the estimates, respectively.

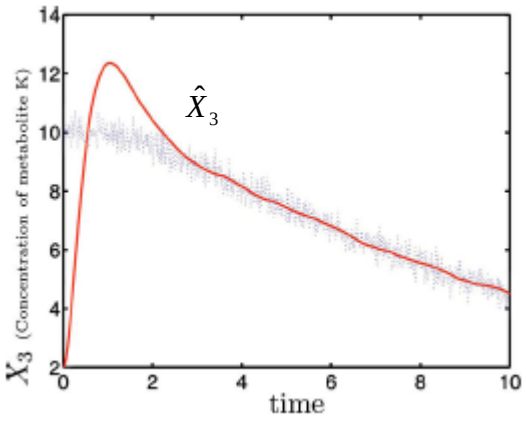


Figure 3.6 Numerical simulation: solid lines represent the filltered states and the dotted lines represent the noisy measured state for the evolution in time of metabolite K concentration.

Notice that the initial bad estimation is due to the initial conditions that have been chosen far away from the real ones. This behavior could be improved with a better knowledge of the initial conditions.

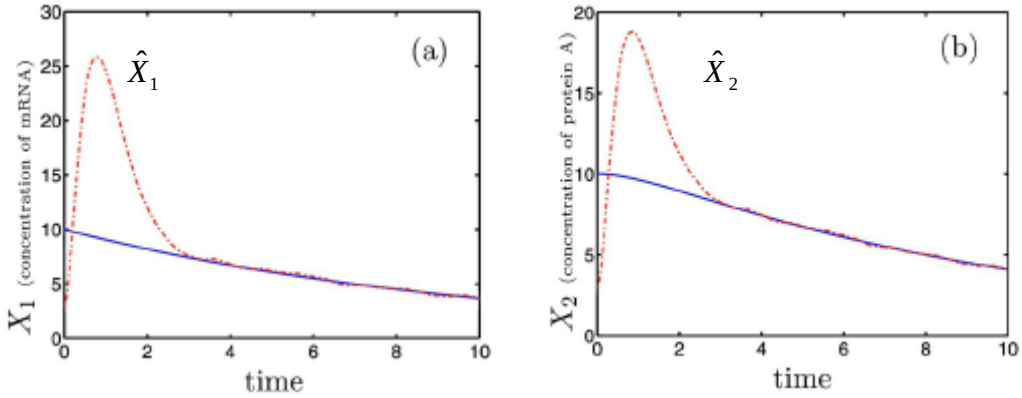


Figure 4.6 Numerical simulation: solid lines represent the true states generated by the original process endowed with the Hill regulatory function and dotted lines represent the estimated concentrations provided by the software sensor without any knowledge about the regulatory function. Plot (a) represents the evolution of mRNA concentration in time and plots (b) the variation of the concentration of protein A in time.

6.5 Three-gene circuit case

In this section we extend the previous results to a more complicated case that can occur in eukaryotic cells. The case corresponds to the coupled regulation of three genes in which the metabolite resulting from the translation of gene 1 becomes the substrate for the synthesis of the metabolite catalyzed by the enzyme translated from gene 2, and similarly for gene 3, but the metabolite 3 becomes the repressor of all the three genes involved, as shown in Fig. **5.6**. In this case the model is given by an extension of the model given by Eqs. (1.6) an (2.6). That results in the following system of differential equations:

$$\frac{d}{dt}[mRNA_1] = K_1 R([Met_3]) - \gamma_1 [mRNA_1]$$

$$\frac{d}{dt}[Enz_1] = K_2 [mRNA_1] - \gamma_2 [ENz_1]$$

$$\frac{d}{dt}[Met_1] = K_3 [Enz_1] - \gamma_3 [Met_1] - \alpha_1 [Enz_2]$$

$$\frac{d}{dt}[mRNA_2] = K_4 R(Met_3) - \gamma_4 [mRNA_2]$$

$$\frac{d}{dt}[Enz_2] = K_5 [mRNA_2] - \gamma_5 [Enz_2]$$

$$\frac{d}{dt}[Met_2] = K_6[Enz_2] - \gamma_6[Met_2] - \alpha_2[Enz_3]$$

$$\frac{d}{dt}[mRNA_3] = K_7 R([Met_3]) - \gamma_7[mRNA_3]$$

$$\frac{d}{dt}[Enz_3] = K_8[mRNA_3] - \gamma_8[Enz_3]$$

$$\frac{d}{dt}[Met_3] = K_9[Enz_3] - \gamma_9[Met_3]$$

where [$mRNA_3$], [Enz_i] and [Met_i] represent the concentration of mRNA, enzymes and metabolites for each gene respectively. We select as the measured variables the metabolites because we want to show that through the measurement of stable molecules such as the metabolites, it is possible to infer the concentration of unstable molecules such as the mRNAs. Note that the equations are coupled through the dynamics of the metabolites. Moreover, we will assume that the dynamics of mRNA is bounded but unknown. As we showed in the previous sections our new system can be written as:

$$\dot{X}_1 = X_2 + d_1, \tag{9.6}$$

$$\dot{X}_2 = K_3 X_3 - \gamma_3 X_2 - \alpha_1 X_8, \tag{10.6}$$

$$\dot{X}_3 = K_2 X_4 - \gamma_2 X_3, \tag{11.6}$$

$$\dot{X}_4 = X_5, \tag{12.6}$$

$$\dot{X}_5 = \phi_1(X), \tag{13.6}$$

$$\dot{X}_6 = X_7 + d_2, \tag{14.6}$$

$$\dot{X}_7 = K_6 X_8 - \gamma_6 X_7 - \alpha_2 X_{13}, \tag{15.6}$$

$$\dot{X}_8 = K_5 X_9 - \gamma_5 X_8, \tag{16.6}$$

$$\dot{X}_9 = X_{10}, \tag{17.6}$$

$$\dot{X}_{10} = \phi_2(X), \tag{18.6}$$

$$\dot{X}_{11} = X_{12} + d_3, \tag{19.6}$$

$$\dot{X}_{12} = K_9 X_{13} - \gamma_9 X_{12}, \tag{20.6}$$

$$\dot{X}_{13} = K_8 X_{14} - \gamma_8 X_{13}, \tag{21.6}$$

$$\dot{X}_{14} = X_{15}, \tag{22.6}$$

$$\dot{X}_{15} = \phi_3(X), \tag{23.6}$$

where $mRNA_1 = \dot{X}_4$, $mRNA_2 = \dot{X}_9$, $mRNA_3 = \dot{X}_{14}$, $Enz_1 = \dot{X}_3$, $Enz_2 = \dot{X}_8$, $Enz_3 = \dot{X}_{13}$, $Met_1 = \dot{X}_2$, $Met_2 = \dot{X}_7$, $Met_3 = \dot{X}_{12}$, d_i represent the noise, $\Phi_i(X)$ stand for the unknown dynamics. In addition, the previous systems can be written in the matricial form as
:

$$\dot{X} = \overline{A}X + \overline{B}(X) + Ed, \qquad X \in R^n$$

$$y = \overline{C}X = (C_1 X^1 ... C_m X^m)^T, \tag{24.6}$$

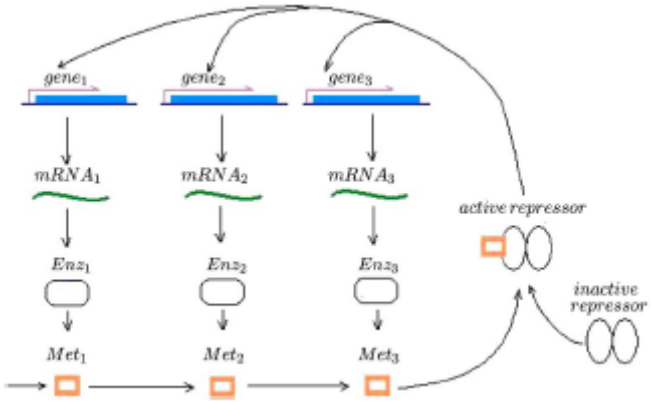


Figure 5.6 The three-gene regulatory circuit under consideration.

According to the scheme presented in the previous section we construct an observer through the following system of deferential equations:

$$\dot{\hat{X}}_1 = \hat{X}_2 + \theta_{11}(X_1 - \hat{X}_1), \tag{25.6}$$

$$\dot{\hat{X}}_2 = K_3 \hat{X}_3 - \gamma_3 \hat{X}_2 - \alpha_1 \hat{X}_8 + \theta_{12}(X_1 - \hat{X}_1), \tag{26.6}$$

$$\dot{\hat{X}}_3 = K_2 \hat{X}_4 - \gamma_2 \hat{X}_3 + \theta_{13}(X_1 - \hat{X}_1), \tag{27.6}$$

$$\dot{\hat{X}}_4 = X_5 + \theta_{14}(X_1 - \hat{X}_1), \tag{28.6}$$

$$\dot{\hat{X}}_5 = \theta_{15}(X_1 - \hat{X}_1), \tag{29.6}$$

$$\dot{\hat{X}}_6 = \hat{X}_7 + \theta_{21}(X_6 - \hat{X}_6), \tag{30.6}$$

$$\dot{\hat{X}}_7 = K_6\hat{X}_8 - \gamma_6\hat{X}_7 - \alpha_2\hat{X}_{13} + \theta_{22}(X_6\hat{X}_6), \tag{31.6}$$

$$\dot{\hat{X}}_8 = K_5\hat{X}_9 - \gamma_5\hat{X}_8 + \theta_{23}(X_6 - \hat{X}_6), \tag{32.6}$$

$$\dot{\hat{X}}_9 = \hat{X}_{10} + \theta_{24}(X_6 - \hat{X}_6), \tag{33.6}$$

$$\dot{\hat{X}}_{10} = +\theta_{25}(X_6 - \hat{X}_6), \tag{34.6}$$

$$\dot{\hat{X}}_{11} = \hat{X}_{12} + \theta\gamma_9\hat{X}_{12} + \theta_{32}(X_{11} - \hat{X}_1), \tag{35.6}$$

$$\dot{\hat{X}}_{13} = K_8\hat{X}_{14} - \gamma_8\hat{X}_{13} + \theta_{33}(X_{11} - \hat{X}_{11}), \tag{36.6}$$

$$\dot{\hat{X}}_{14} = \hat{X}_{15} + \theta_{34}(X_1 - \hat{X}_1), \tag{37.6}$$

$$\dot{\hat{X}}_{15} = \theta_{35}(X_{11} - \hat{X}_{11}), \tag{38.6}$$

where θ_i stand for the observer gain values. Note, that this extension is not a direct application of that developed by Aguilar *et al*. [24], in the sense that this is a extension to the multivariable case. In addition, the matrix A_i is equivalent to a matrix of Brunovsky form, which guarantees the existence, uniqueness and invertibility of the matrix solution S_θ^i [26]. (The existence and the uniqueness of S_θ^i follows from the facts that $-(\theta_i/2)I - A_i$ is of Hurwitz-type and that the pair $(-(\theta_i/2)I - A_i, C_i)$ is observable [27]).

6.6 Conclusion

In this research, a simple software sensor was designed for a schematic gene regulation dynamic process involving end-product inhibition in single gene, operon and three gene circuit cases. This sensor effectively rebuilds the unmeasured concentrations of mRNA and the corresponding enzyme. Thus, the limitation of those experiments in which only the concentration of the catalytically synthesized metabolite is available, can be overcome by employing

the simple software sensor applied here. This is a quite natural case if one takes into account that metabolites are quite stable at the molecular level. At the same time, we can reproduce the concentrations of the unstable molecules of mRNA. This is a difficult task in experiments, despite the fact that the mRNA dynamics has been partially or even totally unspecified.

The same scheme philosophy to build the observer is applied to a three-gene circuit with the purpose to show that the software sensor concept could be in usage in a forward engineering approach. In this research however, we mentioned that we were able to show that the observer scheme designed in [24] for the single output case works well also in a multiple variable case as embodied by a particular genetic circuit given in Fig. 5.6. The most stringent mathematical requirement for this extended applicability to the multiple output case is described below. The linear part of the dynamic system should be a matrix by blocks in which each of the blocks should be of Brunovsky equivalent form. In addition, each subsystem corresponding to a superior block depends only on the subsystem corresponding to the next nearest block. This is a feature similar to the property of Markoff processes.

The Brunovsky equivalent form of the matrix blocks A_i together with the structure of the corresponding output vector C_i generate an observable pair (A_i, C_i), giving us the capability to infer the internal states of the gene network through the knowledge of its external outputs. However, the special Brunovsky equivalent form of the blocks leads to the possible biological interpretation that each block of the linear part of the differential system represents only that contribution of the gene regulation mechanism that comes from reactions occurring in a cascade fashion.

Another important issue that we tackled in this work is related to the way of adding the noise to the output of the dynamic system. Even though this is a typical situation from the standpoint of control process theory, to the best of our knowledge it has not yet been applied in the biological context of gene regulation processes. We stress that this way of including noise effects could have both intrinsic and extrinsic interpretations and therefore assure a more general approach of the noise problems. For example, in phenomenological terms, perturbations on the cells due to the measuring devices and the experimental conditions, together with the noise produced by the nature of the electronic instrumentation, could be equally described in this way.

In addition, this type of nonlinear observer could be used as an online filter being robust with respect to model uncertainties, i.e., neither a known regulation function nor the parameter $K_{1,3}$ is required.

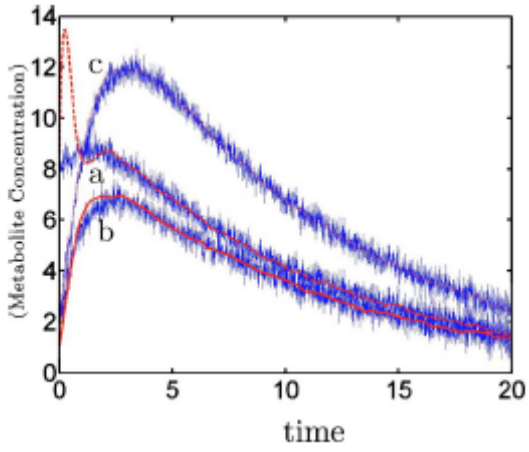


Figure 6.6 Numerical simulation: solid lines represent the filtered states obtained from the noisy measured states for the evolution in time of metabolite concentrations, where a, b and c correspond to metabolite 1, 2, and 3, respectively. The units of the two axes are arbitrary (nondimensional model)

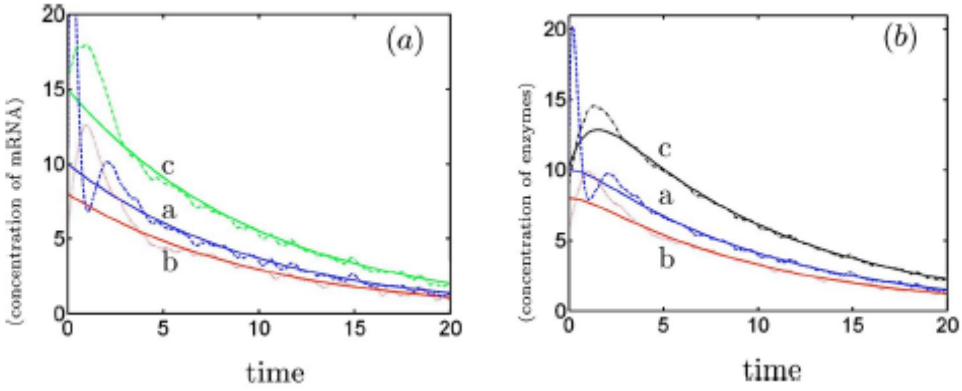


Figure 7.6 Numerical simulation: solid lines represent the true states generated by the original process endowed with the Hill regulatory function and dotted lines represent the estimated concentrations provided by the software sensor without any knowledge about the regulatory function; a, b and c correspond to molecule 1, 2 and 3, respectively. Plot (a) represents the evolution of $mRNA_i$ concentrations in time and plots (b) the variation of the concentration of the corresponding enzymes in time. The axes of the graph have arbitrary units.

References

[1] B. Lewin, Genes VII (Oxford University Press, Oxford, 1999).

[2] See http://www.iee.org/sb

[3] T.M. Yi, Y. Huang, M.I. Simon, J. Doyle, Robust perfect adaptation in bacterial chemotaxis through integral feedback control, *Proc. Natl. Acad. Sci. USA* ***97***, 4649 (2000).

[4] N. Barkai, S. Leibler, Robustness in simple biochemical networks, *Nature* **387**, 913 (1997).

[5] D. G. Luenberger, Observers for multivariable systems, *IEEE Trans. Autom. Control* **11**, 190 (1966).

[6] J.Hasty, J. Pradines, M.Dolnik, and J.J. Collins, Noise-based switches and amplifiers for gene expression, *Proc. Natl. Acad. Sci. U.S.A*. **97**, 2075 (2000); T.Kepler and T. Elston, Stochasticity in transcriptional regulation: origins, consequences, and mathematical representations, *Biophys.J*. **81**, 3116 (2001); P.S. Swain, M.B. Elowitz, and E.D. Siggia, Intrinsic and extrinsic contributions to stochasticity in gene expression, *Proc. Natl. Acad. Sci. U.S.A*. **99**, 12 795 (2002); W.J. Blake, M. Kaern, C.R. Cantor, and J.J. Collins, Noise in eukaryotic gene expression, *Nature (London)* **422**, 633 (2003); F.J. Isaacs, J. Hasty, C.R. Cantor, and J.J. Collins, Prediction and measurement of an autoregulatory genetic module, *Proc. Natl. Acad. Sci. U.S.A.* **100**, 7714 (2003).

[7] E. H. Davidson, J. P. Rast, P. Oliveri, A. Ransick, C. Calestami, C.-H. Yuh, T. Minokawa, G, G. Amore, V. Hinman, C. Clarke, M. I. Arnone, L. Rowen, R. A. Cameron, D. R. McClay, L. Hood, and H. Bolouri, A genomic regulatory network for development, *Science* **295**, 1669 (2002).

[8] T.I. Lee, N.J.Rinaldi, F. Robert, D.T. Odom, Z. Bar-Joseph, G.K. Gerber, N.M. Hannet, C. T. Harbison, C. M.Thompson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B Gordon, B. Ren, J.J. Wyrick, J.-B Tagne, T.L. Volkert, E. Fraenkel, D.K. Gifford, and R.A. Young, Transcriptional regulatory networks in Saccharomyces cerevisiae, *Science* **298**, 799 (2002).

[9] H. De Jong, Modeling and simulation of genetic regulatory systems: A literature review, *J. Comput. Biol*. **9**, 67 (2002)

[10] E. M. Ozbudak, M Thattai, I. Krutser, A.D. Grossman, and A. Van Oudenaarden, Regulation of noise in the expression of a single gene, *Nature Genetics* **31**, 69 (2002).

[11] P.A. Brown and D. Botstein, Exploring the new world of the genome with DNA microarrays, *Nature Genetics* **21**, 33 (1999).

[12] R. J. Lipshutz, S.P. Fodor, T.R. Gingeras, and D.J. Lockhart, High density synthetic oligonucleotide arrays, *Nature Genetics* **21**, 20 (1999).

[13] D.J. Lockhart and E.A. Winzeler, Genomics, gene expression and DNA arrays, *Nature (London)* **405**, 827 (2000).

[14] G. Stephanopoulos, Chemical Process Control (Prentice Hall, Englewood Cliffs, NJ, 1984).

[15] B.C. Goodwin, Temporal Oscillations in Cells (Academic, New York, 1963).

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

[17] D.L. Nelson and M.M. Cox, Lehninger Principles of Biochemistry (Worth, New York, 2000), p. 1077.

[18] M. Santillán and M.C. Mackey, Dynamic regulation of the tryptophan operon: A modeling study and comparison with experimental data, *Proc. Natl. Acad. Sci. U.S.A.* **98**, 1364 (2001).

[19] W.A. Wolovich, Automatic Control Sustems (Saunders Colege, Philadelphia, 1994).

[20] A.J. Krener and A. Isidori, Linearization by output injection and nonlinear observers, *Syst. Control Lett*. **3**, 47 (1983).

[21] J.P. Gauthier, H. Hammouri, and S. Othaman, A simple observer for nonlinear systems: Applications to bioreactors, *IEEE Trans. Autom. Control* **37**, 875 (1992).

[22] J.P. Gauthier and G. Bornard, Observability for any u(t) of a class of nonlinear systems, *IEEE Trans. Autom*. *Control* **AC-26**, 922 (1981).

[23] J.P. Gauthier and I.A.K. Kupka, Observability and observers for nonlinear systems, *SIAM J. Control Optim*. **32**, 975 (1994).

[24] R. Aguilar, R. Martinez-Guerra, and R. Maya-Yescas, State estimation for partially unknown nonlinear systems: a class of integral high gain observers, *IEE Proc.- Control Theory Appl.* **150**, 240 (2003).

[25] R. Aguilar-López, Integral observers for uncertainty estimation in continuous chemical reactors: algebraic-differential approach, *Chem. Eng. J*. **93**, 113 (2003).

[26] R. Martinez-Guerra, R. Suarez, and J. De León-Morales, Asymptotic output tracking of a class of nonlinear systems by means of an observer, *Int. J. Robust Nonlinear Control* **11**, 373 (2001).

[27] R. Hermann and A. J. Krener, Nonlinear controllability and observability, *IEEE Trans. Autom. Control* **AC-22**, 728 (1977).

[28] H. Shim, Y.I. Son, J.H. Seo, Semi-global observer for multi-output nonlinear systems, *Syst. Control Lett*. **42**, 233 (2001).

APPENDIX A: Distance Measures

Euclidian Distance Measure

The Euclidian distance can be considered to be the shortest distance between two points, and is basically the same as Pythagorus' equation when considered in 2 dimensions (Figure 1) .

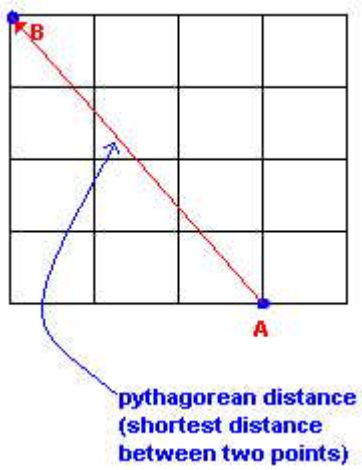


Figure 1. Straight line between A and B.

City Block (Manhattan) Distance

Computationally speaking, this is a cheaper distance measure. Intuitively it is suitable for measurements of discrete data, but it is not restricted to such cases only (see end of Figure 2 legend).

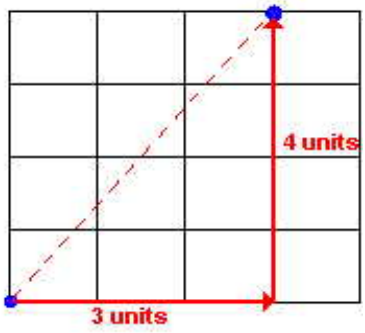


Figure 2. Manhattan distance means to take a 'city walk' to get from one point to another. It does not necessarily have to be the way as portrayed above - it is equally valid to take one unit to the right and one unit up and so on until the destination is reached. As mentioned above, it is intuitively useful for discrete datasets - the intersection between 2 black lines in the picture above could represent one whole unit (e.g. 1 person) and so the 'distance' between the two points protrayed above could be 7 units (e.g. 7 people). It makes sense to use this when one wants the distance to be discrete rather than continuous (e.g. 6.5 people might not be suitable for analysis!)

Chebyshev Distance

If one were to take each element that makes up the two vectors in question, then the two corresponding elements that create the largest difference (i.e. made into positive value) are considered to be the Chebyshev distance. This measure is particularly useful when computation time is absolutely imperative.

Minkowski Distance (MD) of Order *m*

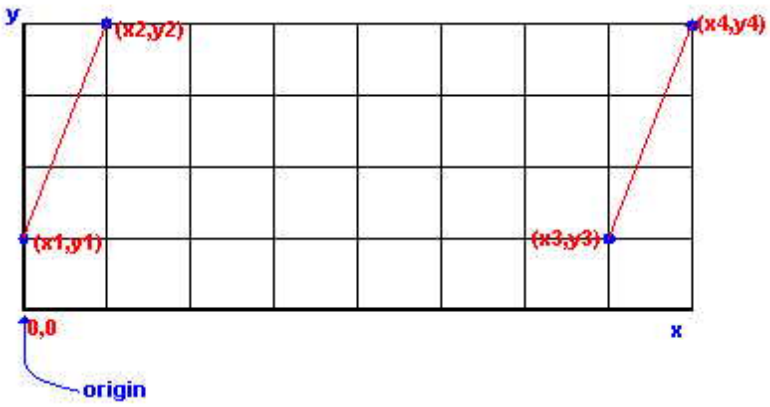
Looking at the equation closely will reveal that when m = 2 the distance metric is exactly the same as as the Euclidian distance. When m = 1, the distance is exactly the same as the City Block distance. As m increases, the metric tends towards a Chebyshev result. Therefore by inceasing m, one can place more numerical value on the largest distance (in terms of elements in the two vectors in question).

The JExpress Manual (available from MolMine) states that the MD measure is the most commonly used distance measure when using ratio scales (when there's an absolute zero). A disadvantage of the Minkowski method is that if one element in the vectors has a wider range than the other elements then that large range may 'dilute' the distances of the small-range elements. The JExpress Manual explains further and suggests other variants of this distance measure to counteract the overpowering effect of wide-range elements.

Quadratic Distance

This is a computationally expensive approach to a weighted distance metric. The weight is defined by Q, a symmetric matrix of correlations.

Canberra Metric



This is usually only meant for non-negative variables only. The Canberra metric makes a summation of a series of ratios between corresponding planar values. This therefore accounts for the distance between two points but also their relation to the'origin'.

Figure 3. The Canberra distance between (x_1,y_1) and (x_2,y_2) is 0.6, whereas for (x_3,y_3) to (x_4,y_4) it is 0.666, even though they are identical in geometric distance. This difference becomes even more apparent in multi-dimensional space. It can therefore be seen that this measure has a bias for distances being measured around the origin. Classification methods very much rely on suitable methods in their decision-making processes to determine whether or not a certain data point belongs to a particular predefined class (that predefinition comes from training).

Pearson Correlation Coefficient

Given fixed positions in *z*-dimensional space, two vectors can be compared by their components with Pearsons Correlation coefficient. The result is always between -1 and 1 inclusive:**1** means there is perfect similarity, i.e. the vectors are identical; **0** means there is no similarity; **-1** means there is perfect dissimilarity, i.e. the vectors are perfectly opposite.

Unlike the Canberra method, if one were to translate both vectors about (by multiplication, addition or subtraction) in *z* space, the same result would be procured because the relation between the two vectors has not changed - in this way, this is similar to the Euclidean method. This metric is therefore independent of the vectors' position in relation to the origin.

Uncentered Pearson Correlation Coefficient

Performing the Pearson Correlation Coefficient metric without taking into account the mean of the components yields an **uncentered** correlation coefficient. The 'normal' Pearsons Correlation Coefficient gives two vectors the value of 1 (perfect similarity) if their shape is identical, even if they are offset from each other. The uncentered flavour of this metric takes into account the relative positions of the vectors with the origin (i.e. their magnitude) and so does not yield 1 in the same situation.

Squared Pearson Correlation Coefficient

This is the same as Pearsons Correlation Coefficient, except that negative values are squared and so are no longer achievable, i.e. one cannot obtain a perfect dissimilarity (-1) anymore. This is extremely useful in gene expression studies - let a vector in this case (gene expression values) represent gene behaviour. Opposite gene expression behaviour would ordinarily be denoted by negative values from Pearsons Correlation Coefficient, but in the Squared Pearsons Correlation Coefficient opposite bahaviours can be made to be synonymous.

Why would this be useful?

If one had an expression profile of gene X with a certain behaviour, and it was found that gene Y was repressed in terms of expression by gene X, then in a Pearsons Correlation Coefficient analysis the value between these two would be -1 (assume there is a 1:1 inverse proportionality). The Squared Pearsons Correlation Coefficient would give +1.

When performing a cluster analysis, using Pearsons Correlation Coefficient as a distance measure would yield clusters such that X and Y would be very far apart from each other - this is because there is no apparent similarity. In fact, biologically speaking, there is a very *close* relationship between X and Y, even though their vectors are opposite. Using the Squared Pearsons Correlation Coefficient would actually reflect this relationship and cluster X and Y very closely.

Calculating Distances of Vectors

Usually one would choose the method which gives the 'best' results in terms of some error function or ability to classify/cluster certain data points. The most commonly used one is the Euclidian distance measure.

The purpose of such measures is to give a numerical value to the amount of dissimilarity between two vectors.

Below are some common measures. The equations below have components x and y, which are the elements of the two vectors in question.

$d_e = \sqrt{\sum_{i=1}^{p}(x_i - y_i)^2}$	Euclidean Distance	-Standard Metric
$d_{cb} = \sqrt{\sum_{i=1}^{p}\lvert x_i - y_i\rvert}$	City Block Distance	-Manhattan Distance -Taxicab Metric
$d_{ch} = \max_i \lvert x_i - y_i\rvert$	Chebyshev Distance	-Minimax Approximation
$d_M = \left\{\sum_{i=1}^{p}(x_i - y_i)^m\right\}^{\frac{1}{m}}$	Minkowski Distance of order m	-L_p Distance
$d_q = \sum_{i=1}^{p}\sum_{j=1}^{p}(x_i - y_i)Q_{ij}(x_j - y_j)$	Quadratic Distance	
$d_{ca} = \sum_{i=1}^{p}\left(\frac{\lvert x_i - y_i\rvert}{x_i + y_i}\right)$	Canberra Distance	
$d_{pcc} = \frac{\sum_{i=1}^{p}(X_i - \bar{X}_i)^2(Y_i - \bar{Y}_i)}{\sum_{i=1}^{p}(X_i - \bar{X})^2\sum_{i=1}^{p}(Y_i - \bar{Y}_i)^2}$	Pearsons Correlation Coefficient	-Linear Correlation Coefficient
$d_{upcc} = \frac{\sum_{i=1}^{p}X_i - Y_i}{\sum_{i=1}^{p}(X_i - \bar{X})^2\sum_{i=1}^{p}(Y_i - \bar{Y}_i)^2}$	Uncentered Pearsons Correlation Coefficient	
$d_{spcc} = \left(\frac{\sum_{i=1}^{p}(X_i - \bar{X}_i)^2(Y_i - \bar{Y}_i)}{\sum_{i=1}^{p}(X_i - \bar{X})^2\sum_{i=1}^{p}(Y_i - \bar{Y}_i)^2}\right)^2$	Squared Pearsons Correlation Coefficient	

The equations for distance measures. In red, most common name and in blue, other commonly used names.

APPENDIX B: Mahalanobis Distance (MD)

Retrieved from "http://en.wikipedia.org/wiki/Mahalanobis_distance"

In statistics, **Mahalanobis distance** is a distance measure introduced by P. C. Mahalanobis in 1936. It is based on correlations between variables by which different patterns can be identified and analysed. It is a useful way of determining *similarity* of an unknown sample set to a known one. It differs from Euclidean distance in that it takes into account the correlations of the data set and is scale-invariant, i.e. not dependent on the scale of measurements.

Formally, the Mahalanobis distance from a group of values with mean $\mu = (\mu_1, \mu_2, \mu_3, \ldots, \mu_p)^T$ and covariance matrix Σ for a multivariate vector $x = (x_1, x_2, x_3, \ldots, x_p)^T$ is defined as:

$$D_M(x) = \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)}.$$

MD can also be defined as dissimilarity measure between two random vectors $\vec{x}$ and $\vec{y}$ of the same distribution with the covariance matrix Σ :

$$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T \Sigma^{-1} (\vec{x} - \vec{y})}.$$

If the covariance matrix is the identity matrix, the MD reduces to the Euclidean distance. If the covariance matrix is diagonal, then the resulting distance measure is called the *normalized Euclidean distance*:

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{p} \frac{(x_i - y_i)^2}{\sigma_i^2}},$$

where σ_i is the standard deviation of the x_i over the sample set.

Intuitive explanation

Consider the problem of estimating the probability that a test point in *N*-dimensional Euclidean space belongs to a set, where we are given sample points that definitely belong to that set. The first step is to find the average or center of mass of the sample points. Intuitively, the closer the point in question is to this center of mass, the more likely it is to belong to the set. However, we also need to know how large the set is.

The simplistic approach is to estimate the standard deviation of the distances of the sample points from the center of mass. If the distance between the test point and the center of mass is less than one standard deviation, then we conclude that it is highly probable that the test point belongs to the set. The further away it is, the more likely that the test point should not be classified as belonging to the set.

This intuitive approach can be made quantitative by defining the normalized distance between the test point and the set to be $\frac{x - \mu}{\sigma}$. By plugging this into the normal distribution we get the probability of the test point belonging to the set.

The drawback of the above approach was that we assumed that the sample points are distributed about the center of mass in a spherical manner. Were the distribution to be decidedly non-spherical, for instance ellipsoidal, then we would expect the probability of the test point belonging to the set to depend not only on the distance from the center of mass, but also on the direction. In those directions where the ellipsoid has a short axis the test point must be closer, while in those where the axis is long the test point can be further away from the center.

Putting this on a mathematical basis, the ellipsoid that best represents the set's probability distribution can be estimated by building the covariance matrix of the samples. The MD is simply the distance of the test point from the center of mass divided by the width of the ellipsoid in the direction of the test point.

Relationship to leverage

MD is closely related to the leverage statistic *h*. The MD of a data point from the centroid of a multivariate data set is (*N* − 1) times the leverage of that data point, where *N* is the number of data points in the set.

Applications

MD is widely used in cluster analysis and other classification techniques. It is closely related to Hotelling's T-square distribution used for multivariate statistical testing.

In order to use the MD to classify a test point as belonging to one of N classes, one first estimates the covariance matrix of each class, usually based on samples known to belong to each class. Then, given a test sample, one computes the MD to each class, and classifies the test point as belonging to that class for which the MD is minimal. Using the probabilistic interpretation given above, this is equivalent to selecting the class with the highest probability.

Also, MD and leverage are often used to detect outliers especially in the development of linear regression models. A point that has a greater MD from the rest of the sample population of points is said to have higher leverage since it has a greater influence on the slope or coefficients of the regression equation.

References

- P.C. Mahalanobis, On the generalised distance in statistics, Proceedings of the National Institute of Science of India 12 (1936) 49-55

APPENDIX C: Diffeomorphism (from Wikipedia)

In mathematics, a **diffeomorphism** is a kind of isomorphism of smooth manifolds. It is an invertible function that maps one differentiable manifold to another, such that both the function and its inverse are smooth.

Definition

Given two manifolds *M* and *N*, a bijective map *f* from *M* to *N* is called a **diffeomorphism** if both

$$f : M \to N$$

and its inverse

$$f^{-1} : N \to M$$

are differentiable (if these functions are *r* times continuously differentiable, *f* is called a C^r-**diffeomorphism**).

Two manifolds *M* and *N* are **diffeomorphic** (symbol being usually $\simeq$) if there is a diffeomorphism *f* from *M* to *N*.

Examples

$$\mathbb{R}/\mathbb{Z} \simeq S^1.$$

That is, the quotient group of the real numbers modulo the integers is again a smooth manifold, which is diffeomorphic to the 1-sphere, usually known as the circle. The diffeomorphism is given by

$$x \mapsto e^{2\pi i x}.$$

This map provides not only a diffeomorphism, but also an isomorphism of Lie groups between the two spaces.

Local description

Model example: if *U* and *V* are two open subsets of $\mathbb{R}^n$, a differentiable map *f* from *U* to *V* is a **diffeomorphism** if

1. it is a bijection,
2. its derivative *Df* is invertible (as the matrix of all $\partial f_i/\partial x_j$, $1 \le i, j \le n$), which means the same as having non-zero Jacobian determinant.

Remarks:

- Condition 2 excludes diffeomorphisms going from dimension *n* to a different dimension *k* (the matrix *Df* would not be square hence certainly not invertible).
- A differentiable bijection is *not* necessarily a diffeomorphism, e.g. $f(x) = x^3$ is not a diffeomorphism from $\mathbb{R}$to itself because its derivative vanishes at 0.
- *f* also happens to be a homeomorphism.

Now, *f* from *M* to *N* is called a **diffeomorphism** if in coordinates charts it satisfies the definition above. More precisely, pick any cover of *M* by compatible coordinate charts, and do the same for *N*. Let φ and ψ be charts on *M* and *N* respectively, with *U* being the image of φ and *V* the image of ψ. Then the conditions says that the map $\psi f \varphi^{-1}$ from *U* to *V* is a diffeomorphism as in the definition above (whenever it makes sense). One has to check that for every couple of charts φ, ψ of two given atlases, but once checked, it will be true for any other compatible chart. Again we see that dimensions have to agree.

Diffeomorphism group

The **diffeomorphism group** of a manifold is the group of all its automorphisms (diffeomorphisms to itself). For dimension greater than or equal to one this is a large group. For a connected manifold *M* the diffeomorphisms act transitively on *M*: this is true locally because it is true in Euclidean space and then a topological argument shows that given any *p* and *q* there is a diffeomorphism taking *p* to *q*. That is, all points of *M* in effect look the same, intrinsically. The same is true for finite configurations of points, so that the diffeomorphism group is *k*- fold multiply transitive for any integer $k \geq 1$, provided the dimension is at least two (it is not true for the case of the circle or real line). This group can be given the structure of an infinite dimensional Lie group, modeled on the space of vector fields on the manifold. In general, this will not be a Banach Lie group, and the exponential map will not be a local diffeomorphism.

Homeomorphism and diffeomorphism

It is easy to find a homeomorphism which is not a diffeomorphism, but it is more difficult to find a pair of homeomorphic manifolds that are not diffeomorphic. In dimensions 1, 2, 3, any pair of homeomorphic smooth manifolds are diffeomorphic. In dimension 4 or greater, examples of homeomorphic but not diffeomorphic pairs have been found. The first such example was constructed by John Milnor in dimension 7, he constructed a smooth 7-dimensional manifold (called now Milnor's sphere) which is homeomorphic to the standard 7-sphere but not diffeomorphic to it. There are in fact 28 oriented diffeomorphism classes of manifolds homeomorphic to the 7-sphere (each of them is a fiber bundle over the 4-sphere with fiber the 3-sphere).

Much more extreme phenomena occur: in the early 1980s, a combination of results due to Fields Medal winners Simon Donaldson and Michael Freedman led to the discoveries that there are uncountably many pairwise non-diffeomorphic open subsets of $\mathbb{R}^4$ each of which is homeomorphic to $\mathbb{R}^4$, and also that there are uncountably many pairwise non-diffeomorphic differentiable manifolds homeomorphic to $\mathbb{R}^4$ which do not embed smoothly in $\mathbb{R}^4$.

APPENDIX D: Lipschitz continuity (from Wikipedia)

This is a *smoothness* condition for functions that is stronger than regular continuity.

A Lipschitz continuous function is limited in how fast it can change. In geometric terms, a line joining any two points on the graph of the function will never have a slope steeper than a certain number called the *Lipschitz constant* of the function.

The concept of Lipschitz continuity can be defined on metric spaces and thus also on normed vector spaces. A generalization of Lipschitz continuity is called Hölder continuity.

Definition

A real valued function f defined on a subset D of the real numbers is called Lipschitz continuous or is said to satisfy a *Lipschitz condition* if there exists a constant $K \geq 0$ such that for all x_1, x_2 in D

$$|f(x_1)-f(x_2)| \leq K\,|x_1-x_2|.$$

The smallest such K is called the Lipschitz constant of the function f.
Alternatively, one can write

$$|f(x_1)-f(x_2)| / |x_1-x_2| \leq K$$

for $x_1 \neq x_2$, i.e., iff the slopes of secants are bounded.

Locally Lipschitz continuous

The function is called *locally Lipschitz continuous* if for every x in D there exists a neighborhood U(x) so that f restricted to U is Lipschitz continuous.

Lipschitz continuity in metric spaces

Given two metric spaces (M,d) and (N,d'), where d and d' denotes the metric on the sets M and N respectively, U is a subset of M, a function

$$f: U \rightarrow N$$

is called *Lipschitz continuous* if there exists a constant $K \geq 0$ such that for all x_1 and x_2 in U

$$d'(f(x_1), (f(x_2)) \leq K\, d(x_1, x_2).$$

The smallest such K is called the *Lipschitz constant* of the function f. If K=1 the function is called *short map*, if K< 1 the function is called *contraction*.

If, for K>1, d' is also bounded below by the metric d, i.e.,

$$K^{-1}\, d(x_1, x_2) \leq d'(f(x_1), (f(x_2)) \leq K\, d(x_1, x_2).$$

then f is called *bi-Lipschitz*.

APPENDIX E: Kalman Filters (KFs)

KFs have been introduced by R. Kalman in 1960 [1]. At the present time, extensive treatments of the KFs can be found in many standard text books [2-4].

On the other hand, the extension of the KFs to nonlinear systems is required for solving many state estimation problems.

Suppose a mathematical model is given for a natural system in the form

$$x_{n+1} = F(x_n, u_n, w_n) \qquad (1)$$

and its measurement (output) function as

$$y_n = H(x_n, u_n, v_n) \qquad (2)$$

where the variables are the state, input, and state noise variables, respectively, in the first equation, and y_n and v_n are the model output and observation noise in the second equation. For continuous systems, x_{n+1} is obtained by numerical integration procedures with x_n, u_n, and w_n as input.

The *optimal* estimate of the state variables x_n in the sense of least mean squares, given the observation of y_n , is the conditional expectation $E(x_n \mid y_n)$. The KF realizes a *recursive procedure* to obtain this conditional expectation for a linear Gaussian system.

However, this optimality cannot be retained for a nonlinear system. Instead, several extension of the linear KF can be used to obtain a *suboptimal* solution for nonlinear systems.

Let $\hat{x}_{n \mid n-1} = x_{\text{prior-est}}$ denote the prior estimation of x_n with its associated covariance matrix as $P_{\hat{x}_{n|n-1}}$

At time n, a new measurement y_n is collected to derive a better estimation of x_n.

One way, which is the optimal one in the case of linear Gaussian systems, is to have a correction term added to $\hat{x}_{n \mid n-1}$ that is based on the difference between the measured y_n and such that

<u>*measurement update*</u>: $x_{\text{pres-est}} = x_{\text{prior-est}} + K_n (y_n - y_{\text{prior-est}})$ (3)

where K_n is termed *Kalman gain*, which, for both linear and nonlinear systems, can be optimally calculated as

$$K_n = P\, x_n\text{-}x_{prior\text{-}est}\,,\; y_n\text{-}y_{prior\text{-}est},\; P^{-1}\; y_n\text{-}y_{prior\text{-}est} \qquad (4)$$

where the first P is the covariance between x_n-$x_{prior\text{-}est}$ and y_n-$y_{prior\text{-}est}$.

The optimality of K_n is a classical result from linear optimal estimation theory.

It is also not difficult to get that the posterior covariance in x is less than the prior covariance in x by a term
of the form

$$(P_{post} - P_{prior})_x = - K_n \quad P_{y_n - y_{prior\text{-}est}} (K_n)^T \qquad (5)$$

The diagonal terms of the posterior covariance give the variances of the posterior estimate of state variables.

Equation (3) is usually called *measurement update* since the upgrade of the prior estimate to a better posterior estimate is achieved with the arrival of a new measurement.

With $x_{\text{pres-est}}$ a prediction of $x_{n+1|n}$ can be made as

$$\underline{\textit{time update}}: \quad \hat{x}_{n+1|n} = E[F(x_{\text{pres-est}} \ldots)] \qquad (6)$$

where E[·] is the expectation operator. It implies the calculation of the conditional probability of x_n on the measurements up to n. This step is usually called *time update*.

The prediction of the measurement can be calculated in the similar fashion as

$$\hat{y}_{n+1|n} = E[H(x_{\text{pres-est}}, u_n, v_n)] \qquad (7)$$

The time update step in the general KF paradigm is essentially the propagation of the expectation and the covariances of random variables through functions.

Nonlinear KFs

Different nonlinear KFs address this propagation problem in different ways [5-8] while the measurement update is conducted in the same fashion. The calculation of K_n is always done at the measurement step of the filtering process according to equation (4). There are many nonlinear filters that follow the basic Kalman filtering structure.

References

[1] R. Kalman,
A new approach to linear filtering and prediction problems,
Trans. ASME, Ser. D, J. Basic Eng., vol. 82, pp. 34-45, (1960).

[2] P. Zarchan and H. Musoff,
Fundamentals of Kalman Filtering: A Practical Approach,
ser. Progress in Astronautics and Aeronautics, Reston, VA: Am. Inst. Aeronautics Astronautics, vol. 190 (2000).

[3] A.H. Sayed,
Fundamentals of Adaptive Filtering,
Piscataway, NJ: IEEE Press/Wiley-Interscience, (2003).

[4] C.K. Chui and G. Chen,
Kalman Filtering: With Real-Time Applications,
ser. Springer Series in Information Sciences, 3rd ed. Berlin, Germany: Springer-Verlag, vol. 17 (1999).

[5] N.J. Gordon, D.J. Salmond, and A.F.M. Smith,
Novel approach to nonlinear/nongaussian Bayesian state estimation,
Inst. Electr. Eng. Proc., F Radar, Signal Process., vol. 140, no. 2, pp. 107-113 (1993).

[6] S. Julier, J. Uhlmann, and H.F. Durrant-Whyte,
A new method for the nonlinear transformation of means and covariances in filters and estimators,
IEEE Trans. Autom. Control, vol. 45, no. 3, pp. 477-482, (2000).

[7] S.J. Julier and J.K. Uhlmann,
Unscenting filtering and nonlinear estimation,
Proc. IEEE, vol. 92, no. 3, pp. 401-422 (2004).

[8] M. Norgaard, N.K. Poulsen, and O. Ravn,
New developments in state estimation for nonlinear systems,
Automatica, vol. 36, no. 11, pp. 1627-1638 (2000).

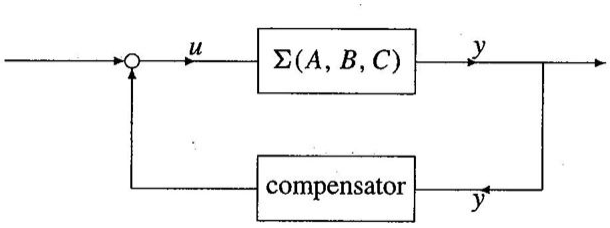
APPENDIX F: Luenberger Observers (LOs)

LOs have been introduced by D. Luenberger in 1963 [1].

In general they occur in problems of stabilizing a dynamical system by state feedback, i.e., *u=Fz.*

This assumes that one can measure the whole state space, which is impossible in realistic cases.

The natural question is how to stabilize the system using only partial information about the state. One answer is to use the measurements (partial information) to estimate the full state (the construction of an observer) and to apply the state feedback on the estimated state.



Consider the state linear system Σ(A,B,C) in the above figure with state space Z, input space *U*, and output space *Y*.

A Luenberger observer for this system is given by

$$\dot{\hat{z}}(t) = A\hat{z}(t) + Bu(t) + L(\hat{y}(t) - y(t))$$
$$\hat{y}(t) = C\hat{z}(t),$$

where $L \in \Lambda(Y, Z)$, the space of bounded linear operators from *Y* to Z. It can be shown that the above LO provides a good estimate of the state *z* provided that system Σ(A,B,C) is exponentially detactable (observable).

LOs for nonlinear systems

M. Zeitz [2] developed an extended LO for nonlinear systems, which is based upon a local linearization technique around the reconstructed state. Other works on the extension of Luenberger observer to nonlinear systems belong to Ciccarela y colaboradores [3] y Kazantzis y Kravaris [4].

References

[1] D.G. Luenberger,
Observing the state of a linear system,
IEEE Trans. Milit. Elctr., vol. 8, pp. 74-80 (1963).

[2] M. Zeitz,
The extended LO for nonlinear systems,
Systems Control Lett., vol. 9, 149 (1987).

[3] G. Ciccarela, M. Dalla Mora, A. Germani,
A Luenberger-like observer for nonlinear systems,
Internat. J Control, vol. 57, 537 (1993).

[4] N. Kazantzis and C. Kravaris,
KA nonlinear Luenberger-type observer with application to catalyst activity estimation, in:
Proc. 1995 American Control Conf., Seattle, Washington, p. 312
(1995).

APPENDIX G: Some References Related to (Maximum) Entropy

[1] C. Predescu,
Entropic effects in large-scale Monte Carlo simulations,
Phys. Rev. E 76, 016704 (2007).

[2] A. Bernacchia,
Continuous or discrete attractors in neural circuits ? A self-organized switch at maximal entropy,
arXiv: 0707.3511v1 (2007).

[3] R. Hanel and S. Thurner,
Generalized Boltzmann factors and the maximum entropy principle: Entropies for complex systems,
Physica A 380, 109 (2007).

[4] S. Olivares and M.G.A. Paris,
Quantum estimation via minimum Kullback entropy principle,
arXiv: 0708.095v1 (2007).

[5] L. Demetrius and T. Manke,
Robustness and network evolution – an entropic principle,
Physica A 346, 682 (2005).

[6] M. Favretti,
Lagrangian submanifolds generated by the maximum entropy principle,
Entropy 7(1), 1-14 (2005).

[7] M. Bauer and D. Bernard,
Maximal entropy random networks with given degree distribution,
arXiv: cond-mat/0206150 (2002).

[8] J. Schneider,
First-order transitions in clustering,
Phys. Rev. E 57, 2449 (1998).

[9] N. Barkai and H. Sompolinsky,
Statistical mechanics of the maximum-likelihood density estimation,
Phys. Rev. E 50, 1766 (1994).

[10] K. Rose, E. Gurewitz, and G.C. Fox,
Statistical mechanics and phase transitions in clustering,
Phys. Rev. Lett. 65, 945 (1990).

Final Conclusions

The published results on the application of the nonlinear observers of Gauthier et al. and Aguilar et al. to the Goodwin biological model of the dynamics of concentrations of biochemical entities within alived cells have been the first interdisciplinary studies in IPICyT.

At the same time, they have been the first publications in the worldwide literature on the application of such observers to gene regulation processes.

On the other hand, in the case of gene expression networks of high complexity the only appropriate treatment of the experimental microarray data are the clustering methods.

In this thesis, we provided a tentative investigation of the promising superparamagnetic type of clustering.