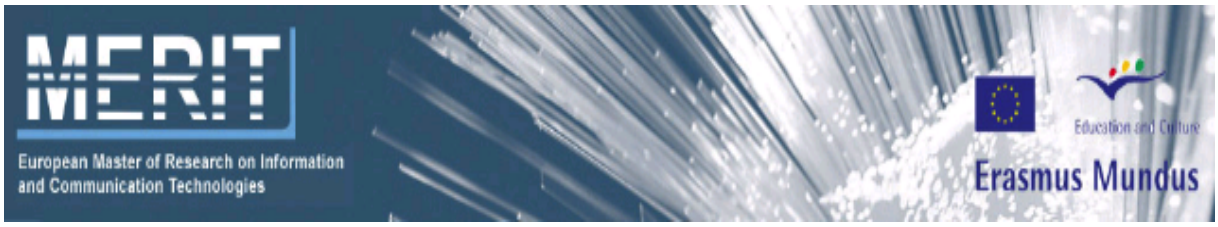



MASTER THESIS

# Study of Digital Signal Processing Tools to Infer Gene Regulatory Networks from Microarrays

AUTHOR: **Pau Bellot Pujalte**
ADVISOR: **Philippe Salembier**

**September 2012**

European Master of Research on Information and Communication Technologies
Signal Theory and Communications Department
Image Processing Group

## Abstract

Since the mid-1990's, the field of genomic signal processing has exploded due to the development of DNA microarray technology, which made possible the measurement of mRNA expression of thousands of genes in parallel. Researchers had developed a vast body of knowledge in classification methods.

The scientific community has developed a broad knowledge of the individual parts involved in the operation of a cell, but we still do not understand how these individual parts interact. For this reason a new type of analysis of the microarray data called Pathways analysis has been developed. This approach considers that genes work together in cascades and do not act for themselves in a biological system. The activity of the genes in a cell is controlled by the gene regulatory networks, which consist of the union and interconnection of the various pathways.

This thesis is placed in the field of computer systems and signal processing applied to biology and aims to study and develop methods to infer the relationship of genes in a large-scale gene network topology where regulation is not known, and must be inferred from experimental data.

First, we present a review and a comparison of the different methods in the state of the art that have tried to solve this challenge with different approaches: Gene networks based in co-expression, Information-theoretic approach, bayesian networks, and finally the one based on differential equations.

Secondly, we present an exhaustive study of two selected techniques, the Z-score and Zavlanos algorithms, in order to analyze their strengths and drawbacks. The chosen methods have been tested on two public datasets: The SOS pathway and a synthetic dataset simulated by computer. The proposed approach obtains good identification results, confirming the goodness of the approach.

And finally, we present an analysis of the ability of the inferred network to predict the behavior of the system to an external perturbation. Also a new approach to boost the identification performance is presented. It is based on an ensemble decision paradigm. It is a preliminary idea but even though, we have found some promising results that demonstrate the potential of the approach.

## Resumen

Desde mediados de los 90, el campo de la genómica fue revolucionado debido al desarrollo de la tecnología de los DNA microarrays, el cual hizo posible la medición de la expresión de mRNA de miles de genes en paralelo. Los investigadores han desarrollado un vasto conocimiento en los métodos de clasificación.

Y aunque la comunidad científica tiene un amplio conocimiento de las distintas partes implicadas en el funcionamiento de una célula, todavía no han logrado entender cómo estas partes individuales interactúan. Por esta razón, un nuevo tipo de análisis de los datos de microarrays llamado análisis de rutas metabólicas se está desarrollando. Este enfoque considera que los genes trabajan conjuntamente y que no actúan por sí mismos en un sistema biológico. La actividad de los genes en una célula está controlada por las redes reguladoras de genes, que consisten en la unión y la interconexión de las diversas rutas metabólicas.

Esta tesis se sitúa en el campo del procesamiento de señal aplicada a la biología y tiene como objetivo estudiar y desarrollar métodos para inferir la relación de los genes en una topología de genes a gran escala donde la regulación es desconocida, y debe ser inferida a partir de datos experimentales.

En primer lugar, se presenta una revisión y una comparación de los diferentes métodos en el estado del arte, que han tratado de resolver este problema con diferentes enfoques: las redes de genes basadas en la co-expresión, la teoría de la información, las redes bayesianas, y finalmente uno basado en ecuaciones diferenciales.

En segundo lugar, se presenta un estudio exhaustivo de las dos técnicas seleccionadas, los algoritmos Z-score y de Zavlanos, con el fin de analizar sus puntos fuertes y débiles. Los métodos elegidos han sido probados en dos conjuntos de datos públicos: El SOS pathway y un conjunto de datos sintéticos simulados por ordenador. El método propuesto permite obtener buenos resultados de identificación, lo que

confirma la bondad del enfoque escogido.

Y, por último, se presenta un análisis de la capacidad para predecir el comportamiento del sistema ante una perturbación externa de la red inferida. Además, se aplica un nuevo enfoque para mejorar la identificación. Está basado en un paradigma de decisión conjunta. Es una idea preliminar, pero a pesar de ello, se han encontrado algunos resultados prometedores que demuestran el potencial de este enfoque.

## Resum

Des de mitjans dels anys 90, el camp de la genòmica va ser revolucionat gràcies al desenvolupament de la tecnologia dels DNA microarrays, la qual va fer possible el mesurament de l'expressió de mRNA de milers de gens en paral·lel. Els investigadors han desenvolupat un vast coneixement en els mètodes de classificació.

I encara que la comunitat científica té un ampli coneixement de les diferents parts implicades en el funcionament d'una cèl·lula, encara no han aconseguit entendre com aquestes parts individuals interactuen. Per això, un nou tipus d'anàlisi de les dades de microarrays anomenat anàlisi de rutes metabòliques s'està desenvolupant. Aquesta tècnica considera que els gens treballen conjuntament i que no actuen per si mateixos a un sistema biològic. L'activitat dels gens en una cèl·lula està controlada per les xarxes reguladores de gens, que consisteixen en la unió i la interconnexió de les diverses rutes metabòliques.

Aquesta tesi es situa en el camp de la processament del senyal aplicat a la biologia i té com a objectiu estudiar i desenvolupar mètodes per inferir la relació dels gens en una topologia de gens a gran escala on la regulació és desconeguda, i ha de ser inferida a partir de dades experimentals.

En primer lloc, es presenta una revisió i una comparació dels diferents mètodes presents a l'estat de l'art, que han tractat de resoldre aquest problema amb diferents enfocaments: les xarxes de gens basats en la co-expressió, la teoria de la informació, les xarxes bayesianes, i finalment un basat en equacions diferencials.

En segon lloc, es presenta un estudi exhaustiu de les dues tècniques seleccionades, els algoritmes Z-score i de Zavlanos, amb la finalitat d'analitzar els seus punts forts i febles. Els mètodes escollits han estat testats amb dos conjunts de dades públiques: El SOS Pathway i un conjunt de dades sintètiques simulades per ordinador. El mètode proposat permet obtindre bons resultats d'identificació, el que confirma la

bondat de la tècnica escollida.

I, finalment, es presenta un anàlisis de la capacitat de predir el comportament del sistema davant d'una pertorbació externa de la xarxa inferida. A més, es presenta una nova técnica per millorar la identificació. Es basa en un paradigma de decisió conjunta. És una idea preliminar, però tot i així, s'han trobat alguns resultats prometedors que demostren el potencial de la idea.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Background

Traditionally, techniques for the study of gene expression were significantly limited in both breadth and efficiency since these studies typically allowed investigators to study only one or a few genes at a time.

However, the DNA microarray technique is a powerful method that provides researchers with the opportunity to analyze the expression patterns of tens of thousands of genes in a short time. Microarray technology is a powerful approach for genomics research. The multi-step, data-intensive nature of this technology has created an unprecedented informatics and analytical challenge.

Microarray technology has become a standard tool in many genomics research laboratories. The reason for this popularity is that microarrays have revolutionized the approach to biological research [1]. Unfortunately, data analyses are often very complex, daunting and confusing.

## 1.2 Motivation of this project

Microarray technology allows researchers to analyze patterns of gene expression with the goal of providing useful information for disease diagnosis or prognosis. These datasets tend to have a large number of gene expression values per sample (several thousands to tens of thousands, even millions), and a relatively small number of samples (e.g., a few dozen samples in relatively rare types of cancer) due to the cost of the experiments. While each sample contains expression information for many genes, it is likely that only a small subset of genes are involved on the

cell status in a specific condition.

However, despite extensive knowledge of individual components, it is not sufficient to understand the functioning a biological cell. In order to gain a systems-level understanding, we also need to examine how the components interact dynamically during operation [2].

The aim of this work is to infer, from gene expression data, the regulatory interactions among genes using computational algorithms. We are interested in the family of those based in the "influence interaction" approach that tries to relate the expression of a gene to the expression of other genes in the cell (gene-to-gene interaction). Although the interaction between two genes in gene networks does not automatically imply a physical interaction between them, it can reflect an indirect regulation via proteins, metabolites embedded that has not been directly measured (see chapter 2).

Moreover, the Department of Signal Theory and Communications (TSC) of the UPC has signed a collaboration agreement with the CELLEX foundation two years ago with the aim to apply Digital Signal Processing tools for analyzing the data generated in the study of cancer. At this moment, this collaboration agreement has two specific points:

- Develop an automatic method to classify each new patient sample in terms of the kind or stage of the disease, with a small margin of error.
- Develop an algorithm to identify the gene regulation networks.

Therefore, the first research line is to analyze the data that come from DNA Microarrays which measure the expression (over or under-expression) of the different genes (around 60000 per patient, depending on the microarray technology), and different kinds of cancer. And the second research line is to infer the gene regulatory networks in order to understand the system, make predictions with the model and bias the classification techniques. This master thesis is embedded in this second research line.

## 1.3 Project goals

The first objective of this project is to review the state of the art techniques and analyze them and choose one that best fit our problem, and then, based on this study, to propose some changes in order to optimize or adapt the technique to this problematic.

The second objective is to study the selected techniques in order to analyze their strengths and drawbacks, and to determine the possible improvements and to select the parameters that provides the best network performance recognition.

Finally, the utility of a network model its ability to make predictions of the outcomes of the system. Therefore the last objective of this project is to make an analysis of the prediction performance of the selected methodology.

## 1.4 Report Structure

This report is divided in 6 chapters, including this first introduction chapter and the last one with the conclusions and future work lines. The remaining chapters follow the chronological order of the project development, from the analysis of the microarray datasets to the application and validation of the proposed algorithms.

Chapter 2 gives a short introduction to the microarray datasets, starting from the most general concepts of biological issues and the construction of microarrays and then focusing on the gene network problematic.

Chapter 3 gives a brief review of the main contributions and different approaches that are proposed by the scientific community, analyzing their strengths and drawbacks. At the end, the main characteristics of the selected algorithms are explained in detail.

In chapter 4 explains the identification performance methodology, then the identification performance results of the different algorithms are presented based on particular gene expression databases.

Chapter 5 discuses about the original contribution of this thesis, it includes two main parts, the prediction analysis of the different algorithms, and a the means to combine the different networks obtained with the different algorithms in order to make more accurate net.

Finally, chapter 6 presents the conclusions reached during this master thesis project and the future work lines.

Additionally, in the appendix A a brief study about some details of the chosen technique is presented.

# Chapter 2

# Genomics review

Genetics is the study of genes. It main goal is to explain what they represent and how they work. In genetics, a feature of a living thing is called a *trait*. Some traits are part of an organism's physical appearance; such as a person's eye-color, height or weight. Other sorts of traits are not easily seen and include blood types or resistance to diseases. The way our genes and environment interact to produce a trait is very complicated. For example, the chances of somebody dying of cancer or heart disease seems to depend on both their genes and their lifestyle.

Genes are made of a long molecule called DNA, which is copied and inherited across generations. DNA is made of simple units that line up in a particular order within this large molecule. The order of these units carries genetic information, similar to how the order of letters on a text carries information. The language used by DNA is called the genetic code, which lets organisms read the information in the genes.

This information corresponds to the instructions for constructing and operating a living organism. Genetic disorders are diseases that are caused by alterations in the genes and are inherited in families. Most of these diseases are inherited in a complex way, with either multiple involved genes, or coming from both genes and the environment. As an example, the risk of breast cancer is 50 times higher in the families most at risk, compared to the families least at risk. Several of the involved genes have been identified but not all of them [3].

## 2.1 Measuring variation in the levels of RNA expression

Genes are the fundamental units of biology. Genes encode proteins, which in turn carry out the processes required for the maintenance of life.

When the genes are active, they produce messages (mRNA), used to provide the information needed to make molecules called proteins in the cells. This process is known as protein synthesis. There is a simple division of labor in the cells: genes give instructions and proteins carry out these instructions, for performing tasks like building a new copy of a cell, doing a cell process or repairing damage. Each type of protein is a specialist that only does one job, so if a cell needs to do something new, it must make a new protein to do this job. Similarly, if a cell needs to do something faster or slower than before, it makes more or less of the corresponding protein. Genes tell cells what to do by telling them which proteins to make and in what amounts [4]. Summing up, during the process of gene expression, the genetic information is first transcribed or copied onto a short-lived messenger RNA molecule. This mRNA is then translated repeatedly into a protein, as specified by the genetic code. This process is often referred to as the 'Central Dogma of Molecular Biology':

$$DNA \longrightarrow mRNA \longrightarrow protein \tag{2.1}$$

Since mRNA is an essential product for synthesizing proteins, the mRNA levels can provide a quantification of gene expression levels. Thus, a gene expression level is thought to be correlated with the approximate number of copies of mRNA produced in a cell.

The aim of the microarrays is to measure the amount of RNA messages in a cell, and therefore to get a rich description of the biology of each cell and each disease. Thus providing an insight into which genes are expressed in a particular cell type, at a particular time, under particular conditions. With this analysis a much richer description of the cell is obtained than using a microscope or an enzyme test, for example.

## 2.2 DNA arrays

Many types of DNA arrays exist. The traditional DNA array is a collection of orderly microscopic *spots*, called features, each with a specific probe attached to a solid surface, such as glass, plastic or silicon biochip. Commonly it is known as a genome chip, DNA chip or gene array,

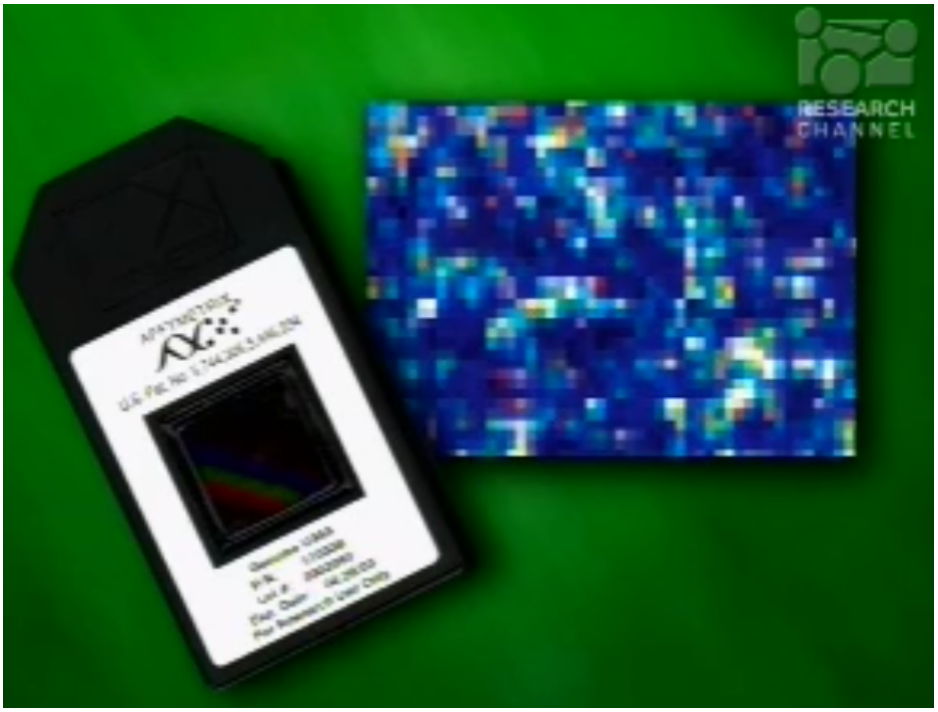
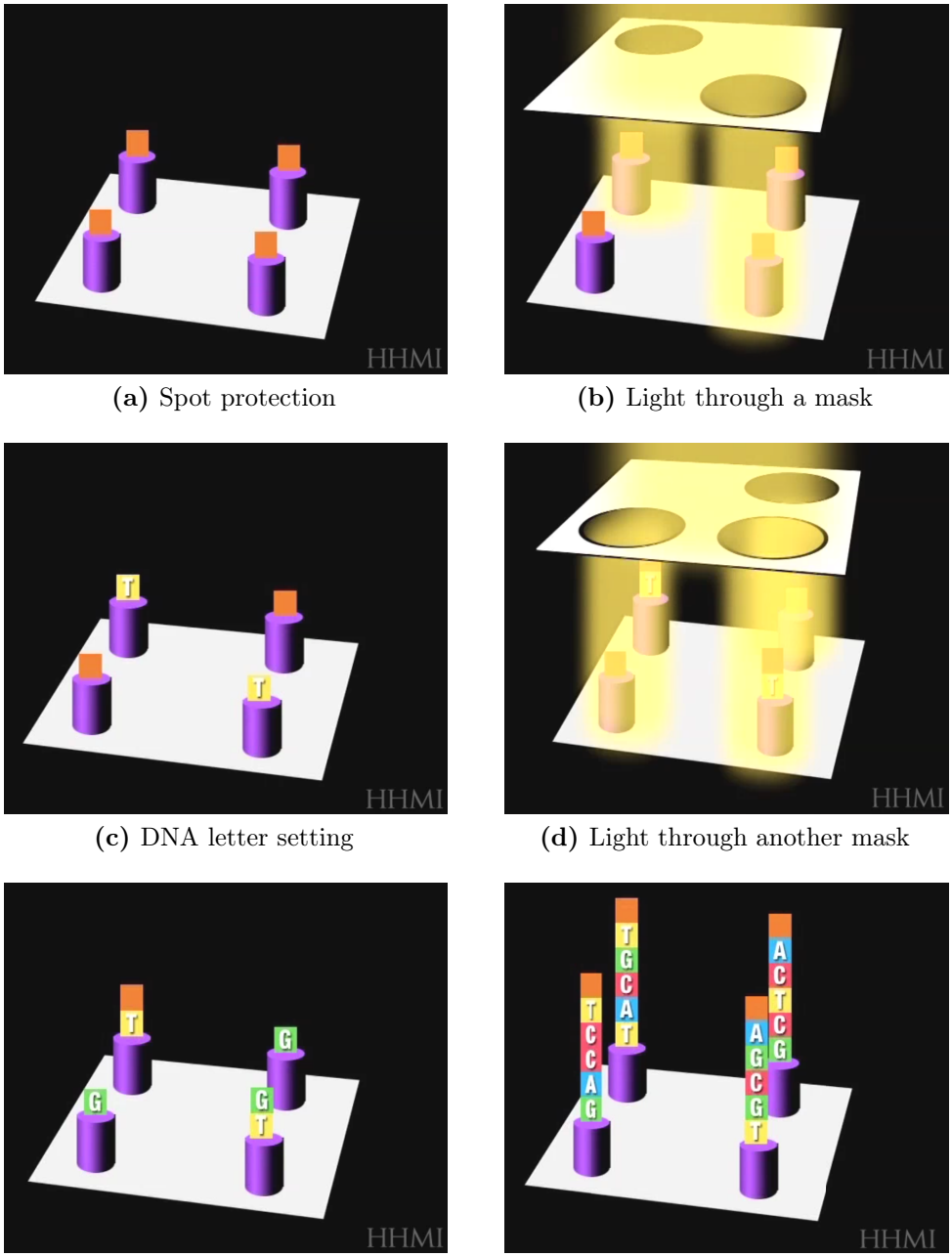



***Figure 2.1.*** *One Affymetrix chip and features example, slide taken from [5].*

which can be observed in Figure 2.1. Thousands of spots can be placed in known locations on a single DNA chip. Each square of the chip has a different DNA sequence, which is a specific 25-letter DNA sequence in every square. Every one of these has whatever DNA sequence is wanted to be specified. This DNA chip will be used to access to the genomic expression of a single sample.

### 2.2.1 Construction of DNA chips

The construction of a biochip is done in the same way as microprocessor chips are built. Every spot is protected (see Figure 2.2a) and a light is shined through a mask (see Figure 2.2b), and the surface is deprotected, then, one of the DNA letters is set (see Figure 2.2c). Then, the surface is re-protected, the light is shined through another mask and deprotect certain spots (see Figure 2.2d) in order to fix the next letter (see Figure 2.2e). After 100 masks it is possible to build up an average of about 25 specific letters in each spot (see Figure 2.2f). In each spot the complementary sequence of every gene in the human genome is built, therefore every spot is a detector for its own gene.




**(a)** Spot protection

**(b)** Light through a mask

**(c)** DNA letter setting

**(d)** Light through another mask

**(e)** Fixation of another DNA letter

**(f)** Results after several DNA letters

***Figure 2.2.*** *Process of construction of a biochip, images taken from [5].*

### 2.2.2 Use of biochips

The first step is to take the RNA from the cell or the tumor, and inject it into the biochip. Each RNA will stick to its own detector, and then with a scanner the intensity of each spot will be read out. This intensity reflects how much each gene is turned on and off (active or not).

Summing up, each one of those chips converts the tumor or cell into a set of gene sequence, which is a long string of data for each patient, telling us which genes are highly expressed or not.

Finally, several gene sequences (biochips) from different patients are grouped in a DNA microarray. The microarrays technique delivers as raw data the information of several patients, stacking all their DNA arrays in a single representation, the various genes are stored in columns and the various patients in rows.

Therefore, with this new technology it has been possible to overcome the traditional techniques for the study of gene expression that were significantly limited in extent and efficiency because it was only possible to study only one or a few genes at a time. With the development of DNA microarray technology it is possible to measure the expression of mRNA of thousands of genes in parallel, and to analyze expression patterns of thousands of genes in a short period of time. As the technology has progressed, microarray has been available to study the expression of an increasingly extensive amount of genes and proteins generated by these genes. With the microarray, it is possible to measure the expression (over or under expression) of the mRNA, with about 54,000 genes in parallel, depending on the platform. However, it has been shown that only a small subset of genes are actually involved in development and physiological functioning of the organism [6].

## 2.3 Understanding the Biology

Thanks to the microarrays the scientific community is building global cancer maps, trying to get the whole expression patterns of RNA variation in a lot of different tumors, identifying the distinguishing set of genes between them.

But although the scientific community has achieved a broad knowledge of the individual parts involved in the operation of a cell, we are far from understanding how cells work and how their operation could be easily handled or managed in the case of the disease [7].

Nearly all the cells of a multicellular organism contain the same DNA, yet this same genetic information yields a large number of different cell types. The fundamental difference between a

neuron and a liver cell, or between a healthy colon cell or a tumoral one, is which genes are expressed. Understanding gene regulation may give important clues about various diseases.

The key question in gene regulation is: What genes are expressed in a certain cell at a certain time? This simple but powerful idea, that genes can be turned on and off, was first proposed by Jacques Monod. His theories remain essentially unchallenged and are the basis of our understanding of gene expression in systems ranging from the simplest viruses to the most complicated multicellular organisms [8].

This fact has motivated a new type of analysis of microarray data, the analysis and discovery of Pathways. This approach considers that the genes work together in cascades and do not act for themselves in a biological system. The pathways therefore reflect a specific cell process which determines its operation, and some disorder could cause cell malfunction (which could lead to cancer development).

**Biochemical Processes** The expression of genes is a tightly regulated process. The physical description is the following: central to this process is a control element known as a promoter – a short stretch of DNA that precedes every gene. The promoter contains a binding site for the RNA polymerase, the protein responsible for transcription. The rate of transcription at a promoter can be increased or decreased by proteins known as transcription factors, therefore the equation 2.1 is completed with this feedback:

$$DNA \overset{\longrightarrow}{\longleftarrow} mRNA \longrightarrow protein \tag{2.2}$$

The additional arrow permits to assemble complex networks of transcriptional and regulatory interactions. A gen can activate multiple genes, and at the same time repress others. The biological encoding of the biochemical process is shown in Figure 2.3.

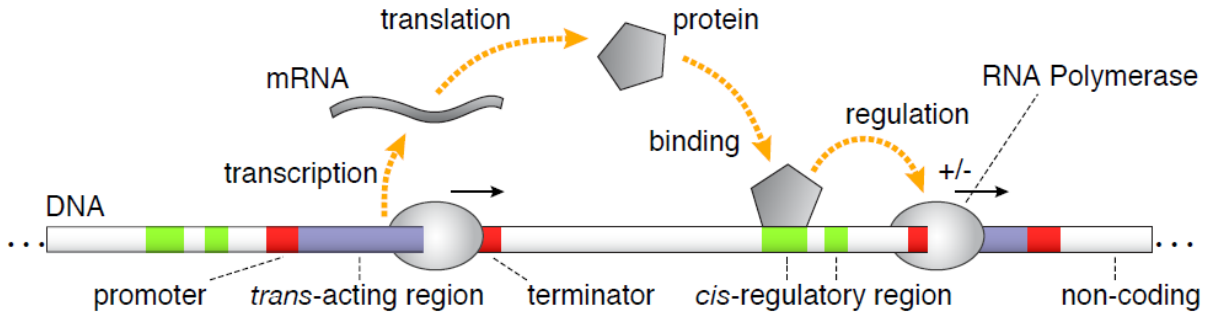



***Figure 2.3.*** *Implicit encoding of genetic interactions in the biological genome. Figure taken from [9].*

So, the amount of genes is not the correct measure of the complexity of an organism. It is the range of possible combinations of expressed genes, which grows exponentially with the gene number.

Here we will not go further into detail of the biochemical and biophysical processes. However excellent textbooks and review articles exist on this topic [10, 11, 12, 13]. That we strongly recommend to the interested reader.

In a level of abstraction, this process is collected by the gene regulatory networks (also known as genetic regulatory networks, or simply gene networks), which are composed of a set of genes, that interact through their RNA and protein products.

Figure 2.4 illustrates this process. (A) Genes are transcribed to mRNA. The transcription rate may be regulated by proteins (incoming arrows). (B) mRNA transcripts are processed and the corresponding proteins are synthesized (translation). Some RNAs have a regulatory function and are not translated. (C) Regulatory proteins (transcription factors) bind to the DNA. Proteins also interact among each other. (D) In practice, a simplified representation is commonly used in gene network models when only mRNA levels are measured (as in microarray data). The real interactions are collapsed in this simplified representation.

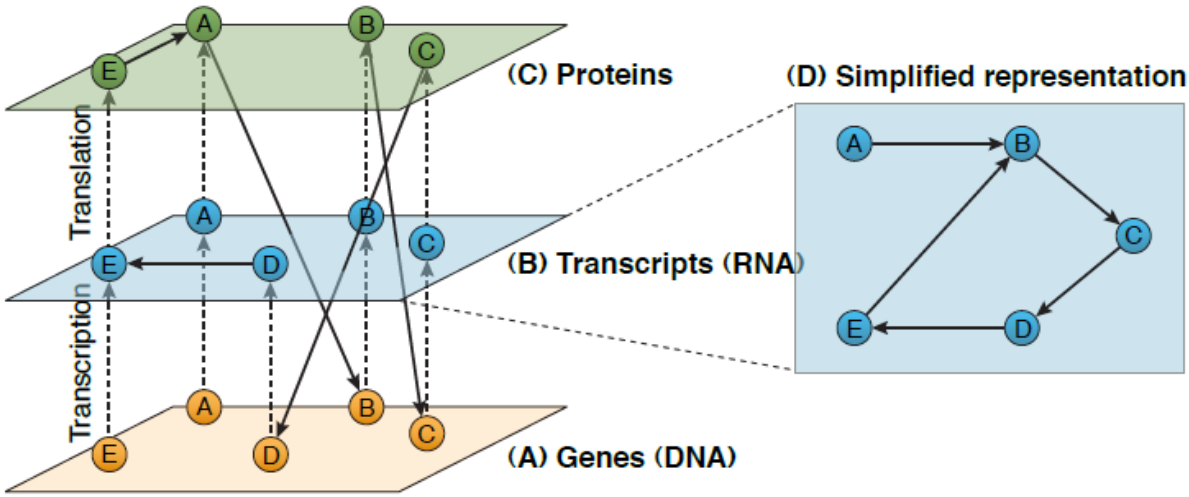



***Figure 2.4.*** *Illustrative example of a gene regulatory network. Figure taken from [9].*

## 2.4 Gene regulatory networks

Gene networks are often represented in simplified form as graphs, where each gene is represented by a single node, and the links are regulatory influences between the genes. In gene-network

reverse engineering (see chapter 3), gene expression levels are often measured in terms of mRNA concentrations. In this case, the nodes are associated with the mRNAs of the genes, and the regulatory influences can be thought of as the "projection" of the different types of regulatory interactions onto the "RNA space", as shown in Figure 2.4 D.

## 2.5 Experiments and resulting types of data

As previously explained, the basic assumption of most reverse engineering algorithms is that causality of transcriptional regulation can be inferred from changes in mRNA expression profiles. We are interested in identifying the regulatory components of the expression of each gene. That is caused because the additional regulation levels, such as proteins and ncRNA concentrations, are currently not available in a sufficient quantity for incorporation in reverse engineering. Therefore, they are neglected or included as hidden factors in diverse gene regulatory models, as in Figure 2.4. Probably this will change in future reverse engineering research.

There are two main kind of datasets:

- Observational: The data contains the system response and the expression of genes under different observations of natural conditions.
- Experimental: The data contains the system response and the expression of genes under different experiments, such as, "over-expression", "perturbation", or "knockdown" experiments. After the perturbation has been performed, cells grow under constant physiological conditions to reach the steady-state. Then, the mRNA concentrations are measured and compared to the concentrations of unperturbed cells, called Wild type, in the same physiological conditions [14].

We are focusing on the second type of data, also called genetic perturbation experiment. There are different kind of experiments:

- Knockouts. Is a single-gene deletion. A knockout is obtained by setting the transcription rate of this gene to zero. Ideally, an independent knockout is provided for every gene of the network.
- Knockdowns. Knockdowns are obtained by reducing the transcription rate of the corresponding gene. Ideally, an independent knockdown is done for every gene of the network.
- Over-expressions. Genes are individually over-expressed.

Many different works (for example [15]) have found that this 'local' perturbation experiments, that is a single gene over-expression or knockdown, seem to be much more informative than 'global' perturbation experiments, that is, over-expressing tens of genes simultaneously or submitting the cells to a strong shock.

Many reverse engineering methods assumes that a large number of different interventions have been done, and that we have access to all of them. Typically, each transcription perturbation corresponds to a specific experiment. However, it is assumed that we deal with controllable networks, where we can perturb each individual gene. This is a reasonable assumption for simple organisms such as *E. coli* or *S. cerevisiae*, and obviously this is also correct with simulated data. But maybe, this is not reasonable for more complex organisms as human beings, and therefore this represents a limitation for these techniques.

# Chapter 3

# Reverse engineering gene networks

Gene expression microarrays produce quantitative data on the cell status in a specific condition. As explained in chapter 1, the aim of this work is to infer, from gene expression data, the regulatory interactions among genes using computational algorithms.

Although the interaction between two genes in gene networks does no automatically imply a physical interaction them, it can reflect an indirect regulation via proteins, metabolites and mRNA. And this networks have practical utility for identifying the subset of genes that regulate each other with multiple (indirect) interactions; so on these gene network, models can be used to predict the response of a network to an external perturbation and to identify the genes directly "hit" by a particular perturbation [16].

This situation is quite usual in the drug discovery process, where one needs to identify these genes that are directly interacting with a compound of interest.

In this section the following notations will be used: The expression measurement of gene $i$ is associated with the random variable $x_i$, the set of expression measurements for all the genes with $D$, and finally the interaction between genes $i$ and $j$ with $a_{ij}$. We are going to work only with steady-state data set $(D)$, which in the gene expression values $x_i$, there are various steady states corresponding to different experiments. Even though time-series data would include more information about the system dynamics, identification is more difficult due to the high computational cost and also the difficulty in obtaining the data.

Depending on the used inference algorithm, the resulting gene network can be either an undirected graph, in which the direction of the interaction is not specified $(a_{ij} = a_{ji})$, or a directed graph specifying the direction of the interaction, that is gene $j$ regulates gene $i$, and not vice versa $(a_{ij} \neq a_{ji})$.

In this chapter, we are going to compare and review different reverse engineering approaches: Gene networks based in co-expression, Information-theoretic approach, bayesian networks, and finally the one based on differential equations.

## 3.1 Co-expression and clustering algorithms

Typically, we have access to a matrix of variables which contains the expression of genes under different conditions, in Figure 3.1 there is an example of these kind of datasets. Therefore, a gene expression profile $x_i$ contains the measurements of a gene expression corresponding to several conditions (that could be natural conditions or as result of the various perturbations described in section 2.5).

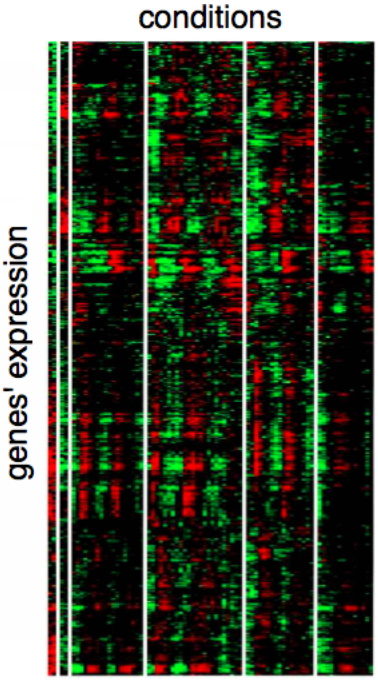



***Figure 3.1.*** *Data matrix measuring the expression of genes under different conditions represented as heat maps to visualize the data.*

For a relatively long time, it has been assumed that similar patterns (in shape) in gene expression profiles under different conditions usually suggest relationships between the genes. Since the coordinated co-expression of genes encodes interacting proteins, studying co-expression patterns can provide insight into the underlying cellular processes. Genes targeted by the same transcription factors tend to show similar expression patterns along time. Clustering, although not properly a network inference algorithm, is currently an important method to visualize and analyze gene expression data. Clustering is based on the idea of grouping genes

with similar expression profiles in clusters [17]. As explained in [18, 19], we have proposed a clustering method to expand the gene set via the generation of metagenes for classification. The metagenes are obtained through hierarchical clustering. Each metagene summarizes in itself a cluster of genes. In order to generate the metagenes set, the Treelet algorithm [20] has been used. As a result the metagenes are linear combinations of genes with common characteristics and so they reduce the noise thanks to the filtering effect of the linear combination.

### 3.1.1 Clustering-based algorithms

The clustering is defined by a grouping similarity specified by a distance metric. One of the most popular distances is the Pearson correlation coefficient among a pair of genes or metagenes that is presented in equation 3.1:

$$r_{ij} = \frac{\sum_{k=1}^{N} x_i(k)x_j(k)}{\sqrt{\sum_{k=1}^{N} x_i^2(k) \cdot \sum_{k=1}^{N} x_j^2(k)}} \tag{3.1}$$

The motivation of the application of the clustering is the belief that coexpressed genes (i.e. genes in the same cluster) have a good probability of being functionally related [17, 21]. But there is a caveat: genes that are in the same cluster could be coexpressed but this fact does not imply the existence of a direct interaction because genes can be related by one or more intermediaries.

The procedure to generate the network is the following one: For a set of $N$ profiles, the most similar pair of genes is used to create a new node in the tree. The algorithm generates a new expression profile that we call a metagene by linear combination. The process is repeated by replacing the two features with this single node, and the process is iterated among the $n-1$ profiles, until only one element remains. Clusters are obtained by pruning the tree at some levels based on a metric. It is assumed that genes in the same cluster regulate each other, that is, each gene represents a node in the network and is connected to all the other genes in the same cluster. So, with this method we can only recover an undirected graph.

### 3.1.2 Z-score

Z-score [22] is one of the simplest inference methods, which is a simplification of conditional correlation analysis. When a regulatory interaction exists between A and B $(A \rightarrow B)$, this

approach assumes that a large expression change in gene B occurs when gene A suffers a knockout or a strong knockdown. Therefore, the z-score for the regulatory interaction $A \rightarrow B$ can be defined with the following equation:

$$Z_{AB} = \frac{x_{B,\Delta A} - \mu_B}{\sigma_B} \tag{3.2}$$

Where $x_{B,\Delta A}$ is the expression value of the gene $B$ in the experiment in which $A$ is perturbed, $\mu_B$ is the mean of the gene B in all the experiments and $\sigma_B$ is the standard deviation of the gene $B$. This definition assumes that $\mu_B$ represents standard expression (if the gene network is sparse [23, 24], most gene interventions would not affect the expression level of gene $B$), also it is assumed that if a cascade of regulation exist i.e $A \rightarrow B \rightarrow C$, the intervention of gene $A$ would produce a larger change in expression of $B$ than the change produced in $C$. The network is recovered by taking the absolute value of z-score and selecting the values that are above of a certain threshold $\gamma$, this is the only parameter to be set.

## 3.2 Information-theoretic approach

This approach uses a generalization of the pairwise correlation coefficient of equation 3.1 that is called Mutual Information (MI) that measures the degree of independence between two genes $x$ and $y$:

$$MI_{x,y} = H_x + H_y - H_{xy} = \sum_{x,y} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)} \tag{3.3}$$

In equation 3.3, $H$ is the usual entropy, defined in equation 3.4 which measures the "information" or "entropy" of a random variable (how unpredictable that variable is).

$$\begin{aligned} H_x &= -\sum_x p(x) \log_2 p(x) \\ H_{xy} &= -\sum_x \sum_y p(x,y) \log_2 p(x,y) \end{aligned} \tag{3.4}$$

For each pair of genes the $MI_{x,y}$ is computed and the edge between genes $i$ and $j$ is set to 0 or 1 depending on a significance threshold [25]. $MI$ is more general than the Pearson correlation coefficient, but in practice $MI$ and Pearson correlation produce almost identical results [26].

Another drawback of this approach is that $MI$ is symmetric, $MI_{x,y} = MI_{y,x}$, therefore the recovered network is an undirected graph.

## 3.3 Bayesian networks

Gene regulatory networks represent the mechanisms that make up the functioning of an organism under given condition. Gene regulatory networks can be recovered using Bayesian networks (BN). These models capture the expression levels and genetic data in discrete variables, related through conditional probability tables capturing regulating and polymorphism effects, including possibly non-linear effects.

A Bayesian network is a graphical model for probabilistic relationships among a set of random variables $X_1, \ldots, X_n$. These relationships are encoded in the structure of a directed acyclic graph (DAG) $G$, whose vertices are the random variables $X_i$. The relationships between the variables are described by a joint probability distribution. In this approach, the genes are "nodes" and the interactions between genes are "edges", associated with the edge is a conditional probability table that provides estimates of the likelihood of the sate of a particular gene given the state of another that regulates it. A Bayesian Network is defined to be a pair $(G, \Theta)$, where $G$ is the DAG and $\Theta$ is the conditional distribution for each variable given its parents $P(X_i|Parents(X_i))$. Bayesian networks only allow dependencies between a node and its parents. A conditional independence statement encoded by the network structure defines the conditional probability distributions; in the case of genes, the factors that influence its expression. A schematic overview of the theory underlying Bayesian networks is given in Figure 3.2.

But learning BNs is computationally expensive, ideally all potential network topologies that correspond to all possible sets of directed acyclic graphs should be assessed and this result in a combinatorial explosion (NP-Complete problem) [27]. For this reason, a general alternative is to perform a heuristic search algorithm, such as greedy hill climbing [28] optimizing some scoring function that reflects how well the BN describes the gene expression data $D$. The score can be defined using the Bayes rule:

$$P(D|G) = \frac{P(G|D) \cdot P(G)}{P(D)} \tag{3.5}$$

where $P(G)$ can reflect some a priori knowledge on the network structure, and $P(D|G)$ is a function to be chosen by the algorithm that evaluates the probability that the data $D$ has been generated by the graph $G$. One of the most popular score is the Bayesian Dirichlet equivalence

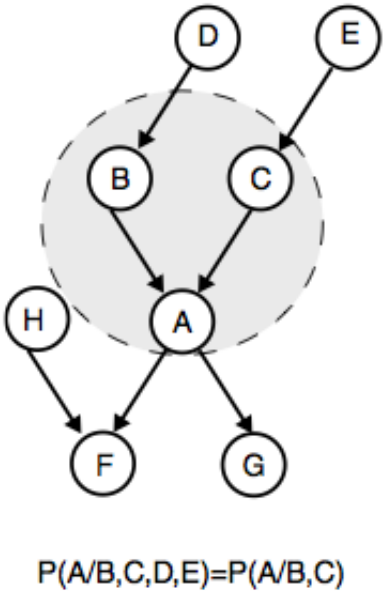



***Figure 3.2.*** *A schematic overview of the theory underlying Bayesian networks. Figure extracted from [15].*

(BDe) [29] that incorporates a penalty for complexity to prevent against overfitting of data.

As it has been commented, the learning problem is difficult and it can even be undetermined, so several high-scoring networks can be recovered. In order to solve this problem, several works have proposed bootstrapping strategies [30]. Bootstrapping is used to select the most probable regulatory interaction and to obtain confidence estimates for the interactions. The problem with these approaches is that they often find local maxima and do not converge to the globally optimal solution. This fact may cause a failure in the aim to find "realistic" networks.

But this can be mitigated with the use of domain-specific knowledge that can provide a useful bias in order to lead the search to near-optimal solutions of a particular problem [31]. A number of possibilities exist to provide prior seeds for biasing the network topology: pathway/interaction databases, networks deduced from the published biomedical literature [32] or Gene Ontology databases. Also, in order to reduce the complexity of the data, a statistical filtering in order to identify the most significant genes thought to be relevant to the system being analyzed, such as Fold Change and t-student test as have been used in [33]. The expression data for these genes are then discretized using a multinomial model [34]. For example, in [35], it is proposed to use three mutually exclusive and exhaustive bins (under-expressed, unchanged, and over-expressed) by equal-width binning.

But the most important limitation of the Bayesian networks is that they cannot contain cycles like feedback loops that are common in genetic regulatory networks. Also, the Bayesian networks only model probabilistic dependencies among variables and not causality that implies

that the parents of a node are not necessarily also the direct causes of its behavior.

## 3.4 Differential equations

This approach [36] is focused on gene external perturbation experiments, such as a treatment with a drug or a genetic perturbation, so it fits with our data study. In contrast with the Bayesian networks (section 3.3) the methods are not based on conditional probabilities and they are deterministic approaches.

As mentioned in chapter 2, a single gene is often regulated by multiple transcription factors which interact with one another. Since these concentrations can themselves change over time due to regulation of the genes, we need to consider the regulatory network as a whole and also tackle with the continuously varying quantities and their rates of change in time, that is, to work with the framework of differential equations.

Therefore, a set of Ordinary Differential equations (ODEs) could describe the gene regulation with one ODE for each gene, as function of other genes:

$$\dot{x}_i(t_k) = f_i\left(x_1, \ldots, x_N, u, \theta_i\right) \tag{3.6}$$

where $\theta_i$ is the set of parameter that describes the interactions between the genes, $u$ is the external perturbation to the system and $\dot{x}_i(t_k) = \mathrm{d}x_i(t_k)/\mathrm{d}t$. The variable $x_i(t_k)$ has been obtained by sampling from a continuous-time signal $x_i$, and then each $k$ value corresponds to a sample. The external perturbation is an experimental treatment that can alter the transcription rate of the genes in the cell. An example of perturbation is the treatment with a chemical compound (i.e. a drug), or a genetic perturbation involving over-expression or knockdown of particular genes [37]. This approach tries to recover the unknown parameters $\theta_i$ starting from a chosen $f_i(\cdot)$ using some optimization technique.

### 3.4.1 Analogy with neural networks

Systems of equations of the general form of 3.6 were exhaustively studied by [38] in order to model neural networks. We think, as in [13], that this field provides a good framework to understand the model and give to the reader an insight on the subject.

In the neural network context, the quantity $x_i$ is the activity of a single neuron, and the function $f_i(\cdot)$ couples neurons to one another across synapses. The neural activity is a continuous

variable, changing continuously over time, analogous to the expression level of a gene.

Early models described neurons as binary units which could perform thresholding operations. In these models, $x_i$ is $0$ or $1$, and neural activity is updated discretely according to the inputs received:

$$x_i(t+1) = \Theta_i \left( \sum_j w_{ij} x_j(t) - \mu_i \right) \tag{3.7}$$

Here, $\Theta(s)$ is a step function, equal to $1$ if $s \geq 0$, and $0$ if the $s < 0$. The weight matrix $w_{ij}$ describes the strength of the interaction between input neuron $j$ and the output of neuron $i$. If the weighted input to neuron $i$ crosses the threshold $\mu_i$, then the neuron is activated.

We can generalize the model and convert the binary activity variables to continuous variables and also overcome the discrete time-steps operation. Essentially, the neurons are assumed to adopt their new activities instantly upon update. Of course, the change of activity might occur gradually, with different neurons, arriving at an equation of the form of 3.6.

### 3.4.2 Working with steady state data

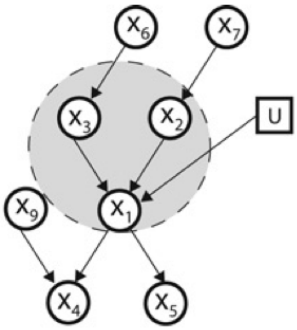
Nonlinear genetic networks as in 3.6 can have multiple stable steady state points, each one typically corresponding to a phenotypical state of the system. Then, the dynamics in a neighborhood of any given equilibrium point can be approximated by a set of linear differential equations. Therefore, the network is described as a function of the concentrations of every other genes in the cell and the external perturbations:

$$\dot{x}_i(t_k) = \sum_{j=1}^{N} a_{ij} x_j(t_k) + b_i u(t_k) \ \forall k = 1 \dots K \tag{3.8}$$

where $K$ is the number of experiments, $\dot{x}_i(t_k)$ reflects the rate of change of concentration of gene $i$ at time $t_k$, as represented in Figure 3.3.

Since we are interested in the case of steady-state data, the system is in a steady state, then the behavior of the system will continue into the future, so the equation 3.8 should become independent of time, that is $\dot{x}_i(t_k) = 0$. Therefore the equation is then simplified:

$$\sum_{j=1}^{N} a_{ij} x_j = -b_i u \tag{3.9}$$




$$\mathrm{d}x_1(t)/\mathrm{d}t = a_{11}x_1(t) + a_{12}x_2(t) + a_{13}x_3(t) + bU(t)$$

$$\Downarrow$$

$$x_1(t_k) = a_{11}\int_0^{t_k} x_1(t)\,\mathrm{d}t + a_{12}\int_0^{t_k} x_2(t)\,\mathrm{d}t + a_{13}\int_0^{t_k} x_3(t)\,\mathrm{d}t + b\int_0^{t_k} U(t)\,\mathrm{d}t$$

***Figure 3.3.*** *Gene regulation model. Figure taken from [37].*

This equation is used by the Network Identification by multiple Regression (NIR) algorithm [39]. The method only needs the information of the perturbed genes in each experiment and a sparseness assumption which determines a maximum number of genes that a single gene can be regulated by. The equation 3.9 is solved as a classic linear-regression problem [40].

In order to find the minimal model the NIR algorithm instead of involving a combinatorial exploration of all possible network topologies, the algorithm introduces an a priori limitation on the connectivity of the network. And it performs a combinatorial search on this limited set. But although the set is reduced, this process is still computationally very hard. Some works have proposed instead an heuristic search algorithms, to perform an exploration to the "state space" of the problem in an attempt to optimize some scoring function. The problem with these approaches is that they often find local maxima and do not converge to the globally optimal solution.

### Zavlanos algorithms

Other algorithm also based in a set of differential equations is proposed in [41]. As previously, if equilibrium is assumed $\tilde{x} = \dot{x} - x_{eq} = 0$, and if the perturbation $u$ is sufficiently small and constant, then the system will stabilize at a new equilibrium that can be formulated by the following set of linear equations:

$$A\tilde{x} + Bu = 0 \tag{3.10}$$

Where $A \in \mathbb{R}^{N\times N}$ encodes the interactions between the $N$ individual genes at the given equilibrium, and the matrix $B \in \mathbb{R}^{N\times p}$ specifies the affected genes by the perturbation. Typically, each transcription perturbation corresponds to a different experiment, and two new matrices could be defined in order to contain all the $M$ steady-state mRNA concentrations $\tilde{X} = [\tilde{x}_1 \dots \tilde{x}_M] \in \mathbb{R}^{N\times M}$ and the corresponding perturbations $U = [u_1 \dots u_M] \in \mathbb{R}^{p\times M}$. In order to model the nonlinearity and the noise in the measurements $(\Delta X)$ that could be different from the linear model, a residual of the linear model as $\eta \triangleq A\Delta X$ is defined:

$$A(\tilde{X} + \Delta X) + BU = [AX + BU = 0] = \eta \tag{3.11}$$

Once we have written this formulation, we can find the linear model that best fit the experiments by minimizing $\eta$ in some norm. So, the network identification problem can be stated in the following words (extracted from [39]):

*Given a steady-state transcription perturbation and mRNA concentration data $U$ and $X$, determine the sparsest stable matrix $A$ that results in sufficiently small residual $\eta$, while incorporating any a priori biological knowledge regarding the presence, absence, or nature of specific gene interactions.*

The matrix $A$ is assumed to be sparse due to the sparsity of the biological networks [23, 24] and also the stability condition is quite reasonable from a biological point of view and necessary for observability of the steady-state [41]. This problem can be formulated as an optimization problem, which satisfies some constraints:

$$\begin{aligned} &minimize\ t \cdot card(A) + (1-t)\epsilon \\ &subject\ to\ \|AX + BU\|_1 \leq \epsilon, A \in S, \epsilon > 0 \end{aligned} \tag{3.12}$$

Where $card(A)$ is the cardinality of the matrix, it measures the "number of elements different from zero", and specifies the sparsity of the matrix. $\|A\|_1$ denotes the $l_1$ norm of the matrix $A$, however any other norm could be used [41], in section A.2 there is an analysis of the usage of the euclidian norm instead of $l_1$ nom.

The parameter $t$ is used in order to control the existent compromise between the sparsity of the matrix $A$ and a best fit to the data. The matrices $X$,$B$,$U$ are the problem data, while $S = s_{ij} \in \{0, +, -, ?\}^{N\times N}$ encodes the a priori knowledge about the network in the form of a partial sign pattern between all pairwise genes in the networks which can be set by means of linear constrictions: positive interactions $(+)$, negative interactions $(-)$, no interaction at all

$(0)$, or no a priori information $(?)$.

$$A \in S \Leftrightarrow \begin{cases} a_{ij} \geq 0 & \text{if } s_{ij} = + \\ a_{ij} \leq 0 & \text{if } s_{ij} = - \\ a_{ij} = 0 & \text{if } s_{ij} = 0 \\ a_{ij} \in \mathbb{R} & \text{if } s_{ij} = ? \end{cases} \tag{3.13}$$

In order to solve the problem avoiding the combinatorial hard nature of the problem Convex Optimization is used [42]. This discipline studies the problem of minimizing convex functions over convex sets. The convexity property can make optimization in some sense "easier" than the general case - for example, any local minimum must be a global minimum.

Therefore a convex relaxation of the cardinality cost function should be done. It is possible to use a weighted $l_1$ relaxation of the cardinality constraint [43], which leads to much more scalable linear constraints, resulting in the following convex problem:

$$\begin{aligned} &minimize\ t \sum_{i \neq j} w_{ij}|a_{ij}| + (1-t)\epsilon \\ &subject\ to\ \|AX + BU\|_1 \leq \epsilon, A \in S, \epsilon > 0 \end{aligned} \tag{3.14}$$

So the variables in the problem are the matrix $A$ and the fitting error $\epsilon$. If the cardinality cost function is relaxed in order to be convex, then by means of an iterative procedure based on the solution of linear programs (convex optimization methods) [42, 44], we will be able to identify the matrix that best fits possibly noisy network data while satisfying the a priori information. The algorithm that simply intends to solve equation 3.14 will be called Unstable (as it does not guarantee stability of the network).

As mentioned previously, in [41], it is proposed to incorporate a stability condition. This restriction makes sense in the biological field. A system of differential equations is stable if all the eigenvalues $\lambda_i$ of $A$ lie in the left-half plane [45], $Re\{\lambda_i\} < 0$.

The stability of the matrix A as a linear constraint can be included relying on a theorem of Gersgorin disc for estimating eigenvalues [46]: Let $A \in \mathbb{R}^{n \times n}$ be a $n \times n$ matrix, with entries $a_{ij}$. For $i \in \{1, \ldots, n\}$ we can define the deleted absolute row sums of $A$ by $R_i(A) \triangleq \sum_{i \neq j} |a_{ij}|$. Then, all eigenvalues of $A$ are located in the union of $n$ discs:

$$G(A) \triangleq \bigcup_{i=1}^{n} \left\{ z \in \mathbb{C} \middle| |z - a_{ii}| \leq R_i(A) \right\} \tag{3.15}$$

Since $A$ and $A^T$ have the same eigenvalues, it is possible to obtain a similar Gersgorin disc theorem for the columns in $A$. If we require that

$$a_{ii} \leq -\sum_{j \neq i} |a_{ij}|, \text{ for all } i = 1, \ldots n \tag{3.16}$$

Then all discs $\left\{ z \in \mathbb{C} \middle| |z - a_{ii}| \leq R_i(A) \right\}$ would lie in the left half plane $\mathbb{C}_-$ and therefore all the eigenvalues of $A$ would be in this plane implying that $A$ is stable. However, these constraints also impose strict structural constraints on the entries of $A$. Particularly, all the diagonal entries of $A$ should be non-positive and also the matrix $A$ may result as a diagonally dominant matrix.

Although this constraint can be relaxed applying a similarity transform on $A$ in order to make it stable but not necessarily diagonally dominant:

$$G(V^{-1}AV) \triangleq \bigcup_{i=1}^{n} \left\{ z \in \mathbb{C} \middle| |z - a_{ii}| \leq \frac{-1}{v_i} \sum_{i \neq j} v_j |a_{ij}| \right\} \tag{3.17}$$

As in equation 3.16 we can require that $a_{ii} \leq -1/v_i \sum_{j \neq i} v_j |a_{ij}|$, for all $i = 1, \ldots n$ and this constraints could be introduced as a set of linear inequalities that can be incorporated in equation 3.14 resulting in the following optimization problem:

$$\begin{aligned} &minimize\ t \sum_{i \neq j} w_{ij} |a_{ij}| + (1-t)\epsilon \\ &subject\ to\ \|AX + BU\|_1 \leq \epsilon, A \in S, \epsilon > 0 \\ &\qquad a_{ii} \leq \frac{-1}{v_i} \sum_{j \neq i} v_j |a_{ij}|, \text{ for all } i = 1, \ldots n \end{aligned} \tag{3.18}$$

The weights $v_i$ are updated iteratively penalizing the Gersgorin discs that lie in the left half plane and at the same time should break the diagonal dominance in the associated row. As a result of this process the convergence of the resulting algorithm is slower than the one without this constraints (equation 3.14). Another problem described in [41] is that for certain ill-conditioned problem instances, the algorithm may find periodic solutions. From now on the

algorithm based in equation 3.18 will be named as Gersgorin.

In order to tackle the stable constraint there is another solution consisting in a semi-definite approximation. It is based in the original algorithm, which allows $A$ to be unstable see equation 3.14. But after some "small" perturbations $D$, the matrix $A$ can be converted into a stable matrix, $A' = A + D$, and at the same time the desired sign pattern and sparsity structure is maintained. This is performed thanks to a constraint that ensures stability if a symmetric positive definite Lyapunov matrix $P$ fulfills the following condition [47]:

$$(A + D)^T P + P(A + D) \prec 0 \ ^1 \tag{3.19}$$

This set of constraints could be incorporated as generalized inequalities in equation 3.14 via a semi-definite formulation, resulting in the following optimization problem:

$$\begin{aligned} &minimize\ t \sum_{i \neq j} w_{ij}|a_{ij}| + (1-t)\epsilon \\ &subject\ to\ \|AX + BU\|_1 \leq \epsilon, A \in S, \epsilon > 0 \\ &A^T P + PA \prec 0 \end{aligned} \tag{3.20}$$

From now on, the algorithm based on equation 3.20 will be named SDP (semi-definite programming).

## 3.5 Conclusions on the different approaches

### Co-expression and clustering algorithms

The motivation of the application of the clustering is the belief that coexpressed genes (i.e. genes in the same cluster) have a good probability of being functionally related. In the networks recovered through "correlation", the genes whose expression is most strongly correlated over a set of conditions are linked, these methods can find true and/or known relationships, as well as many new ones.

But the genes that are in the same cluster could be coexpressed but this fact does not imply

[1] $\prec$ is used to denote generalized inequality with respect to the positive semidefinite cone $\mathbf{S}^n_+$, see [42]

the existence of a direct interaction because genes can be related by one or more intermediaries. So, the main drawback of this technique is that it is assumed that the genes in the same cluster regulate each other. This implies than, each gene is connected to all the other genes in the same cluster. Also, with this method we can only recover an undirected graph, and with unclear meaning. Moreover, the networks are not predictive. What would happen if a determined gene $i$ is deleted?

### Bayesian networks

The most important limitation of the Bayesian networks is that cannot contain cycles like feedback loops that are common in genetic regulatory networks. Also, the Bayesian networks only model probabilistic dependencies among variables and not causality that implies that the parents of a node are not necessarily also the direct causes of its behavior.

### Differential equations

Since this approach is focused on gene external perturbation experiments, such as a treatment with a drug or a genetic perturbation, we think it is the most suited one for our problem, so it fits with our data study. In contrast with Bayesian networks (section 3.3) these methods are not based on conditional probabilities and they are deterministic approaches.

### Chosen approach

After having reviewed these different algorithms and approaches to infer gene networks, we think that the approach which best fits with our problem is the one based on differential equations. Between all different methods that works inside this field, we have choose the one proposed by Zavlanos et al [41] because it can include biological information and avoid heuristic searches. We will also study the z-score method [22] due to its simplicity and because for small networks it has achieved one of the best performances in the DREAM (Dialogue on Reverse-Engineering Assessment and Methods) challenges [48].

# Chapter 4

# Experimental results of performance identification

In this chapter we analyze the efficiency of the selected approaches by studying networks for which the experimental data as well as the ground truth is available. The studied datasets consist on artificial noisy data sets as well as on real experimental dataset in a known subnetwork of the SOS pathway, provided in [39].

With datasets that have a known network we could evaluate the performance of the different algorithms, being able to measure the false positive, negative, etc.

## 4.1 Performance identification measures

In order to evaluate the four different algorithms and perform a study to determine the optimum value of tradeoff $t$ of the algorithms of Zavlanos [41], we will calculate the sensitivity, specificity, and precision, which are defined in equation 4.1. Since these algorithms take advantage of the a priori knowledge about the network these metrics are only calculated in the unknown interactions. As has been explained previously, we consider that the gene regulatory models have only three discrete states (1: activation, -1: inhibition, 0: no-regulation), therefore we can distinguish the following results: TR: True regulation; TZ: True zero; FR: False regulation; FZ: False zero; FI: False interaction; that are collected in a confusion matrix:

Sensitivity ($sen$) is the fraction of the number of found true regulations to all regulations in the model. Specificity ($spe$) is the fraction of correctly found no-interactions to all no-interactions in the model. Additionally we can calculate as well precision ($pre$) that is the fraction of the number of correctly found regulations to all found regulations in the result.

***Table 4.1.*** *Confusion matrix with definitions. TR: True regulation; TZ: True zero; FR: False regulation; FZ: False zero; FI: False interaction;*

| | | Calculated network | | |
|---|---|---|---|---|
| | | 1 | 0 | -1 |
| True network | 1 | TR | FZ | FI |
| | 0 | FR | TZ | FR |
| | -1 | FI | FZ | TR |

$$\begin{aligned} sen &\triangleq \frac{TR}{TR+FZ+FI} \\ spe &\triangleq \frac{TZ}{TZ+FR} \\ pre &\triangleq \frac{TR}{TR+FR+FI} \end{aligned} \tag{4.1}$$

To have a visual measure of performance, we use the Receiver Operating Characteristic (ROC) curve. This curve plots the sensitivity of the identification results against $1 - specificity$. The best possible identification, will give a point in the upper left corner of the plot, representing $100\%$ of sensitivity, and $100\%$ specificity. A completely random guess will give a point along the diagonal line (line of no discrimination).

Plots of the four results identification examples are shown in the ROC space in the Figure 4.1. The result of method A clearly shows the best identification among A, B, and C, because lies above the diagonal line and is the closest to the upper left corner. The result of B lies on the random guess line (the diagonal line), and therefore its accuracy is 50%. The C method has negative identification power. The C' would perform the best because is the closest result to the upper left corner.

In order to obtain a single value measure for one result we can use the distance from the perfect identification as defined in the following equation:

$$d(sen, spe) \triangleq \sqrt{(1-sen)^2 + (1-spe)^2} \tag{4.2}$$

This distance measure $d$ combines the sensitivity and specificity equally weighted to a single value measure. Low values indicate good reconstruction performances. This distance is incorporated in the ROC curves as slashed blue lines that join points with a constant distance with perfect reconstruction.

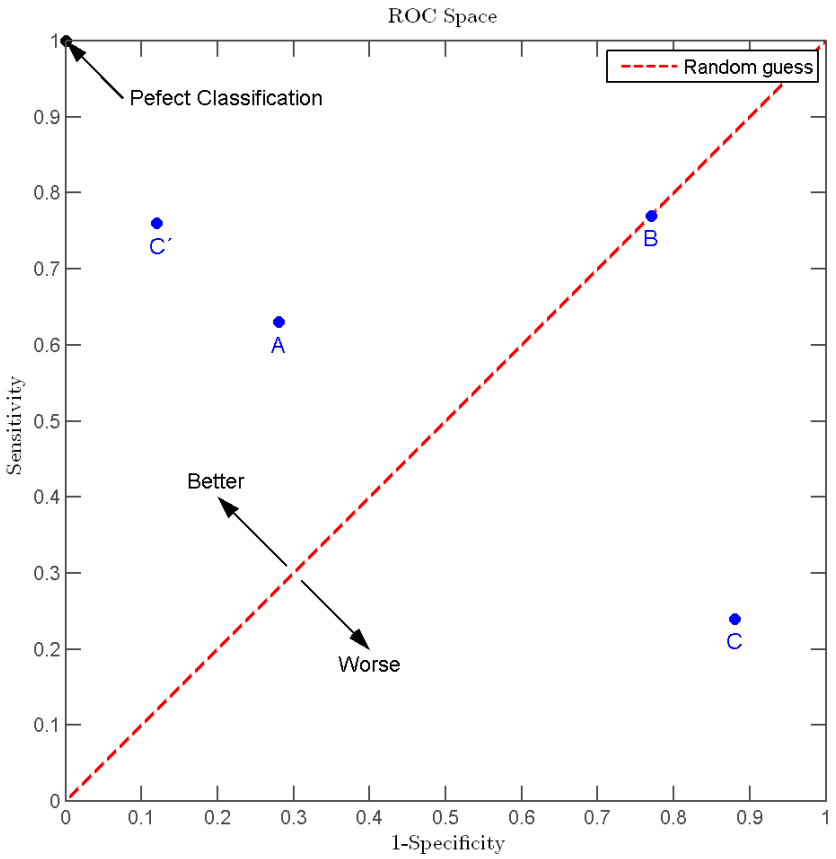



***Figure 4.1.*** *The ROC space and plots of the four identification examples. Figure taken from [49].*

To exemplify all these concepts in the Figure 4.2 there is a toy example of identification outcome extracted from [50], in Figure 4.2 (a). In this figure, the values for sensitivity, specificity and precision are shown in Figure 4.2(b) along with the plot in the ROC Space marked with a red cross.

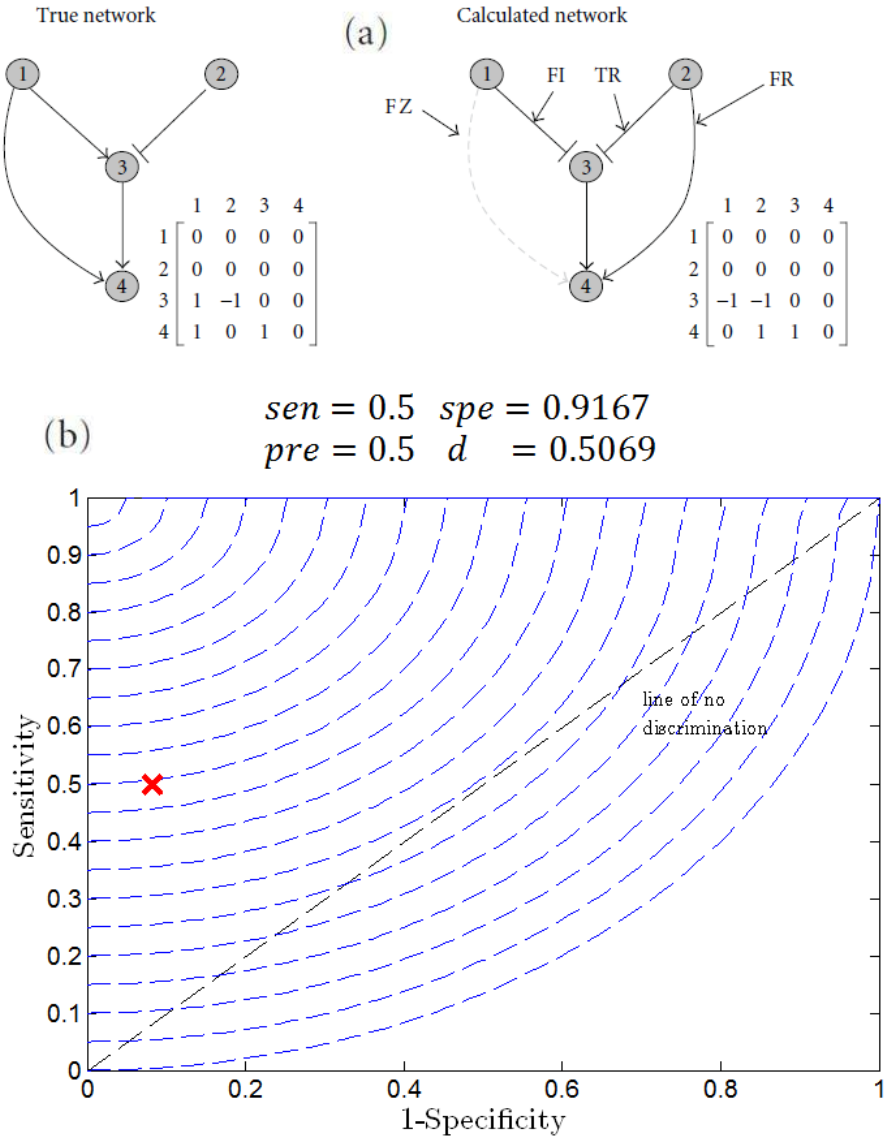



***Figure 4.2.*** *Example of a true network (left) and a calculated network (right). The adjacency matrix represents the network structure. Figure extracted from [50] (b) Values for sensitivity, specificity and precision of the calculated network; and a plotting in the ROC Space marked with a red cross.*

## 4.2 Performance analysis of the Algorithms

### 4.2.1 The SOS Pathway

As mentioned previously, it is a well-known subnetwork that consists of nine genes and several transcription factors and metabolites shown in Figure 4.3, which is extracted from [39]. This subnetwork has a 60% of connectivity that is a value much denser than typical biological networks. The main pathway featured in this network is the pathway between the single-stranded DNA (ssDNA) and the protein LexA that acts as a repressor to several other genes (recA, ssb, dinI, umuDC, and rpoD).

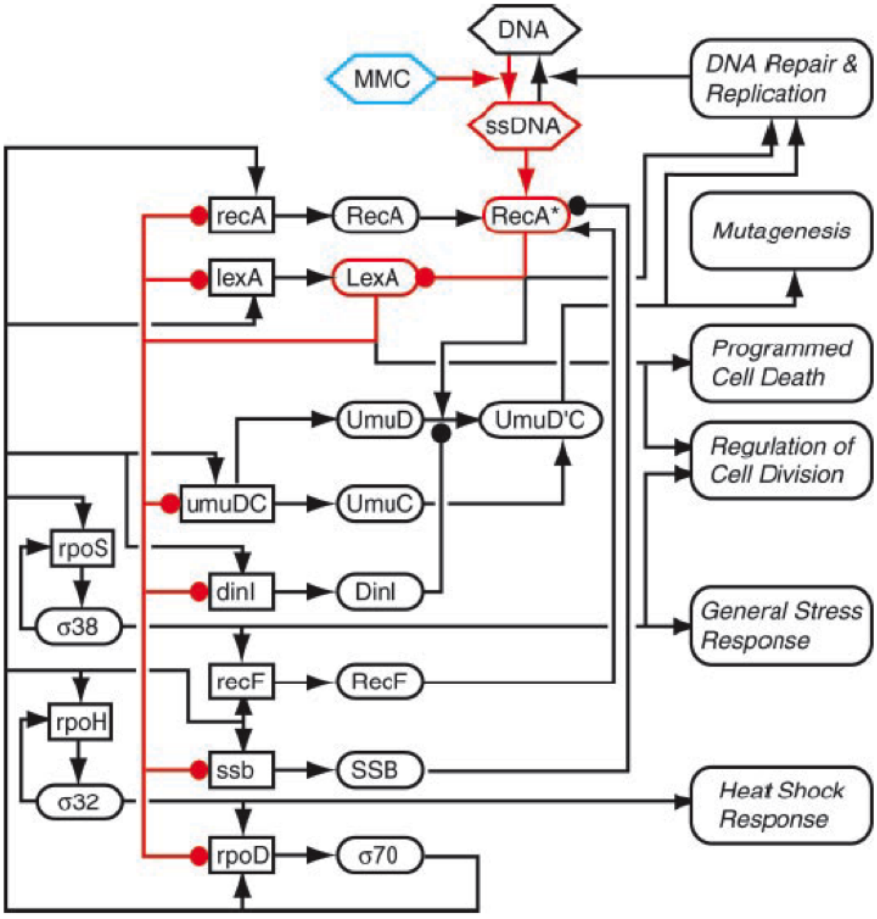



***Figure 4.3.*** *Diagram of interactions in the SOS network. Figure extracted from [39]*

The adjacency matrix produced by this diagram can be observed in Table 4.2.

***Table 4.2.*** *Known regulatory interactions in the SOS test network, extracted from [39], +, −, or 0 indicates a positive, negative, or no regulatory input from the gene in the column to the gene in the row.*

| | recA | lexA | ssb | recF | dinI | umuDC | rpoD | rpoH | rpoS |
|---|---|---|---|---|---|---|---|---|---|
| recA | + | − | − | + | + | − | + | 0 | 0 |
| lexA | + | − | − | + | + | − | + | 0 | 0 |
| ssb | + | − | − | + | + | − | + | 0 | 0 |
| recF | 0 | 0 | 0 | 0 | 0 | 0 | + | 0 | + |
| dinI | + | − | − | + | + | − | + | 0 | 0 |
| umuDC | + | − | − | + | + | − | + | 0 | 0 |
| rpoD | + | − | − | + | + | − | + | + | 0 |
| rpoH | 0 | 0 | 0 | 0 | 0 | 0 | + | + | 0 |
| rpoS | 0 | 0 | 0 | 0 | 0 | 0 | + | 0 | + |

### Evaluation of Z-score

As has been explained previously, the Z-score recovers a regulatory interaction if the measured expression level of a given gene is below $\mu - \gamma\sigma$ or above $\mu + \gamma\sigma$ . In the first case we could assume that there exist an inhibitory regulation and in the second case an enhancing regulation. Table 4.4 shows the performance results for different values of the threshold $\gamma$. These results are also shown in Figure 4.4.

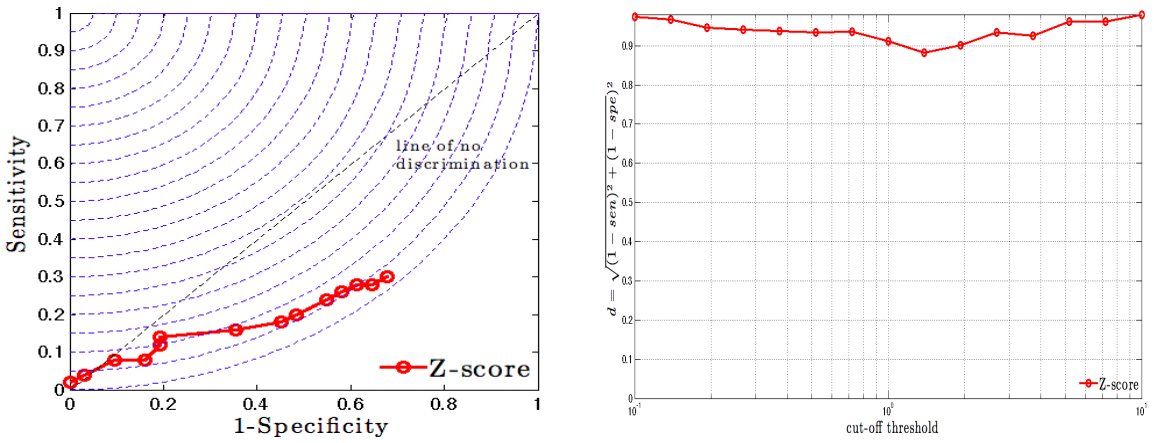



***Figure 4.4.*** *(a) ROC plots of Z-score algorithm for the SOS pathway and different values of the threshold $\gamma$. (b) distance measure $d$.*

***Table 4.3.*** *Performance results of Z-score algorithm for the SOS pathway and different values of the threshold $\gamma$ between 0.1 and 10, in a logarithmically spaced form.*

| $\gamma$ | $sen$ | $spe$ | $pre$ | $d$ |
|---|---|---|---|---|
| 0.1 | 0.3 | 0.323 | 0.231 | 0.974 |
| 0.139 | 0.28 | 0.355 | 0.222 | 0.967 |
| 0.193 | 0.28 | 0.387 | 0.226 | 0.946 |
| 0.268 | 0.26 | 0.419 | 0.228 | 0.941 |
| 0.373 | 0.24 | 0.452 | 0.235 | 0.937 |
| 0.518 | 0.2 | 0.516 | 0.238 | 0.935 |
| 0.72 | 0.18 | 0.548 | 0.237 | 0.936 |
| 1 | 0.16 | 0.645 | 0.25 | 0.912 |
| 1.389 | 0.14 | 0.806 | 0.28 | 0.882 |
| 1.931 | 0.12 | 0.806 | 0.273 | 0.901 |
| 2.683 | 0.08 | 0.839 | 0.25 | 0.934 |
| 3.728 | 0.08 | 0.903 | 0.308 | 0.925 |
| 5.179 | 0.04 | 0.968 | 0.286 | 0.961 |
| 7.197 | 0.04 | 0.968 | 0.286 | 0.961 |
| 10 | 0.02 | 1 | 0.25 | 0.98 |

As could be observed in Figure 4.4, the recovered networks are situated below the diagonal line in the ROC space, therefore these are poor results (worse than random). Our hypothesis to explain the consistently recovering of a poor predictor is that the algorithm is unable of recovering the sign patterns.

In order to prove this hypothesis, the Z-score algorithm will be used to only determine if a regulation exists and not the sign of this regulation. This algorithm will be referred as |Z-score| and since we only care about the topology of the network, the performance indicators of equation 4.1 will be simplified to the following ones:

$$\begin{aligned} sen &\triangleq \frac{TR}{TR+FZ} \\ spe &\triangleq \frac{TZ}{TZ+FR} \end{aligned} \tag{4.3}$$

Table 4.4 shows the performance results of |Z-score| for different values of the threshold $\gamma$.

These results are also shown in Figure 4.5. In this case, the predictions are better than a random guessing and therefore confirms that the algorithm is only well suited for recovering the topology of the network. In the case of SOS pathway the best prediction result is obtained with $\gamma = 1.389$, that is an interaction is recovered if $1.389 <= |\frac{x_{B,\Delta A} - \mu_B}{\sigma_B}|$.

***Table 4.4.*** *Performance results of |Z-score| algorithm for the SOS pathway and different values of the threshold* $\gamma$ *between* $0.1$ *and* $10$*, in a logarithmically spaced form.*

| $\gamma$ | *sen* | *spe* | *pre* | *d* |
|---|---|---|---|---|
| 0.1 | 0.88 | 0.323 | 0.677 | 0.688 |
| 0.139 | 0.86 | 0.355 | 0.683 | 0.66 |
| 0.193 | 0.86 | 0.387 | 0.694 | 0.629 |
| 0.268 | 0.78 | 0.419 | 0.684 | 0.621 |
| 0.373 | 0.68 | 0.452 | 0.667 | 0.635 |
| 0.518 | 0.54 | 0.516 | 0.643 | 0.668 |
| 0.72 | 0.48 | 0.548 | 0.632 | 0.689 |
| 1 | 0.42 | 0.645 | 0.656 | 0.680 |
| 1.389 | 0.38 | 0.806 | 0.76 | 0.65 |
| 1.931 | 0.32 | 0.806 | 0.727 | 0.707 |
| 2.683 | 0.22 | 0.839 | 0.688 | 0.797 |
| 3.728 | 0.2 | 0.903 | 0.769 | 0.806 |
| 5.179 | 0.12 | 0.968 | 0.857 | 0.881 |
| 7.197 | 0.12 | 0.968 | 0.857 | 0.881 |
| 10 | 0.08 | 1 | 1 | 0.92 |

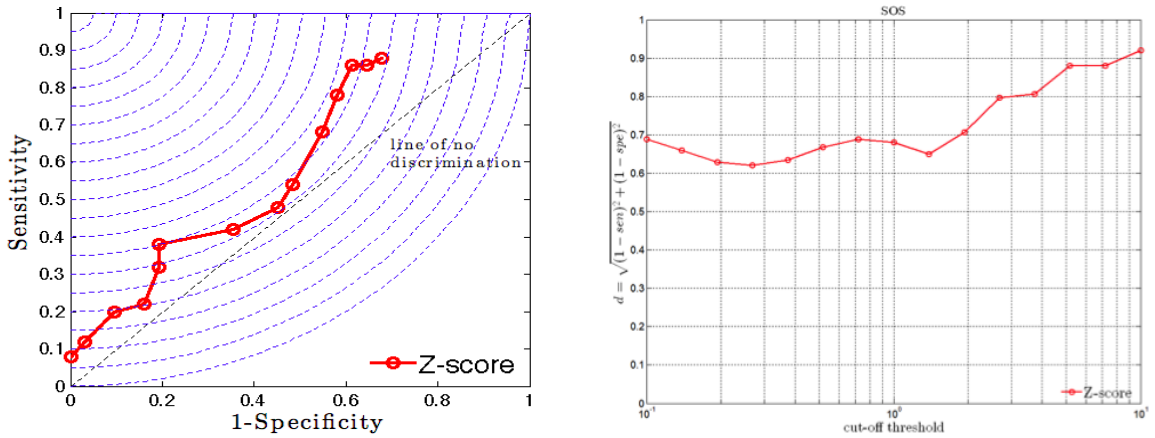



***Figure 4.5.*** *(a) ROC plots of |Z-score| algorithm for the SOS pathway and different values of the threshold $\gamma$. (b) distance measure $d$.*

### Evaluation of Unstable, Gersgorin and SDP algorithms

As mentioned before, the three different Zavlanos algorithms take advantage of the a priori knowledge about the network in the form of a partial sign pattern between all pairwise genes in the networks (matrix $S$ of equation 3.13). The a priori knowledge is depicted in Table 4.2 and has been obtained based on the diagram of Figure 4.3. In order to evaluate the performance of the algorithms not all the a priori knowledge is used and the amount of the considered unknown interactions is changed accordingly for the purpose of identification and performance evaluation. Three different amounts of a priori knowledge have been evaluated: 5, 10 and 19 known interactions over the total of 81 total interactions.

All algorithms were implemented in MATLAB using the "cvx toolbox" for convex optimization problems [44].

### Results with 5 known interactions

In this first subsection, the a priori information is reduced up to 5 interactions that are shown in Table 4.5.

The performance of the three algorithms is evaluated with equations 4.1 and 4.2, that is taking into account the sign of the identified interaction. Table 4.6 shows the results of identifications of the three algorithms evaluated in the 76 "unknown" interactions denoted with a ? in Table 4.5.

***Table 4.5.*** *5 Know interactions given to the algorithms.*

| | recA | lexA | ssb | recF | dinI | umuDC | rpoD | rpoH | rpoS |
|---|---|---|---|---|---|---|---|---|---|
| recA | ? | – | ? | ? | ? | ? | ? | ? | ? |
| lexA | ? | – | ? | ? | ? | ? | ? | ? | ? |
| ssb | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| recF | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| dinI | + | ? | ? | ? | ? | ? | + | ? | ? |
| umuDC | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| rpoD | ? | – | ? | ? | ? | ? | ? | ? | ? |
| rpoH | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| rpoS | ? | ? | ? | ? | ? | ? | ? | ? | ? |

In the Table 4.6 it can be observed that the best performance is given by the Gersgorin Algorithm when $t \leq 0.01$. Some of the results of the Unstable algorithm and the SDP fall under the main diagonal in the ROC space at Figure 4.6a. This effect could be caused by the poor prior information given to the algorithm, and in the case of the SDP by some artificial interactions introduced by the Lyapunov stability condition constraints.

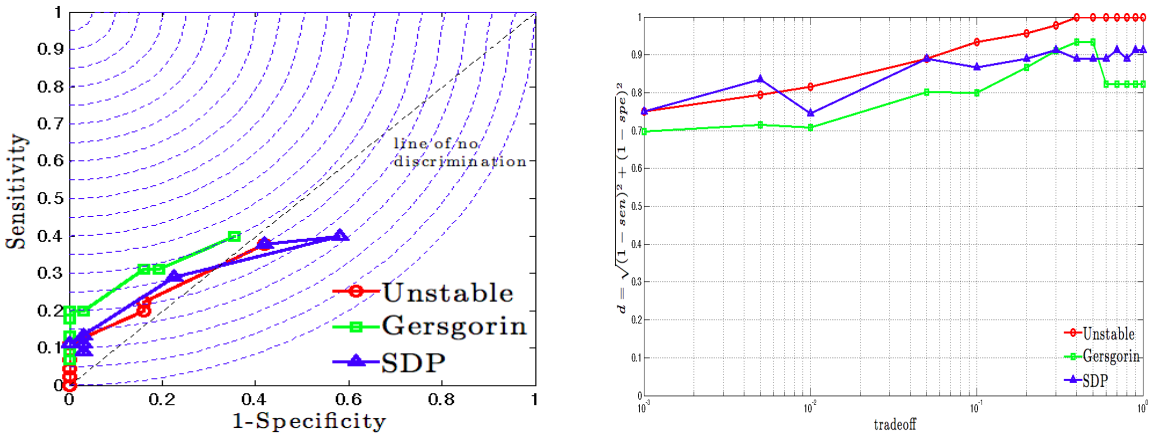



***Figure 4.6.*** *(a) ROC plots of algorithms Unstable (in red), Gersgorin (in green) and SDP (in blue) for the SOS pathway with 5 known interactions and different values of the parameter t. (b) distance measure d of the algorithms.*

***Table 4.6.*** *Performance metrics of algorithms Unstable, Gersgorin and SDP for the SOS pathway with 5 known edges and different values of the parameter t.*

| | Unstable | | | | Gersgorin | | | | SDP | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **t** | **sen** | **spe** | **pre** | **d** | **sen** | **spe** | **pre** | **d** | **sen** | **spe** | **pre** | **d** |
| **0** | 0.378 | 0.581 | 0.5 | 0.75 | 0.4 | 0.645 | 0.621 | 0.697 | 0.378 | 0.581 | 0.5 | 0.75 |
| **0.001** | 0.378 | 0.581 | 0.5 | 0.75 | 0.4 | 0.645 | 0.621 | 0.697 | 0.378 | 0.581 | 0.5 | 0.75 |
| **0.005** | 0.222 | 0.839 | 0.667 | 0.794 | 0.311 | 0.806 | 0.7 | 0.716 | 0.4 | 0.419 | 0.474 | 0.835 |
| **0.01** | 0.2 | 0.839 | 0.643 | 0.816 | 0.311 | 0.839 | 0.737 | 0.708 | 0.289 | 0.774 | 0.65 | 0.746 |
| **0.05** | 0.111 | 1 | 1 | 0.889 | 0.2 | 0.968 | 0.9 | 0.801 | 0.111 | 1 | 1 | 0.889 |
| **0.1** | 0.067 | 1 | 1 | 0.933 | 0.2 | 1 | 1 | 0.8 | 0.133 | 0.968 | 0.857 | 0.867 |
| **0.2** | 0.044 | 1 | 1 | 0.956 | 0.133 | 1 | 1 | 0.867 | 0.111 | 0.968 | 0.833 | 0.889 |
| **0.3** | 0.022 | 1 | 1 | 0.978 | 0.089 | 1 | 1 | 0.911 | 0.089 | 0.968 | 0.8 | 0.912 |
| **0.4** | 0 | 1 | 0 | 1 | 0.067 | 1 | 1 | 0.933 | 0.111 | 0.968 | 0.833 | 0.889 |
| **0.5** | 0 | 1 | 0 | 1 | 0.067 | 1 | 1 | 0.933 | 0.111 | 0.968 | 0.833 | 0.889 |
| **0.6** | 0 | 1 | 0 | 1 | 0.178 | 1 | 1 | 0.822 | 0.111 | 0.968 | 0.833 | 0.889 |
| **0.7** | 0 | 1 | 0 | 1 | 0.178 | 1 | 1 | 0.822 | 0.089 | 0.968 | 0.8 | 0.912 |
| **0.8** | 0 | 1 | 0 | 1 | 0.178 | 1 | 1 | 0.822 | 0.111 | 0.968 | 0.833 | 0.889 |
| **0.9** | 0 | 1 | 0 | 1 | 0.178 | 1 | 1 | 0.822 | 0.089 | 0.968 | 0.8 | 0.912 |
| **1** | 0 | 1 | 0 | 1 | 0.178 | 1 | 1 | 0.822 | 0.089 | 0.968 | 0.8 | 0.912 |

In order to give a visual interpretation, a sign evaluation of the best algorithm of the Table 4.6 corresponding to the Gersgorin algorithm result with $t = 0.001$ is shown in the Figure 4.7. In this figure every pixel represents the interaction between two genes, green color indicates true recognition, red an error in the prediction and yellow indicates a prior known interaction witch are not evaluated.

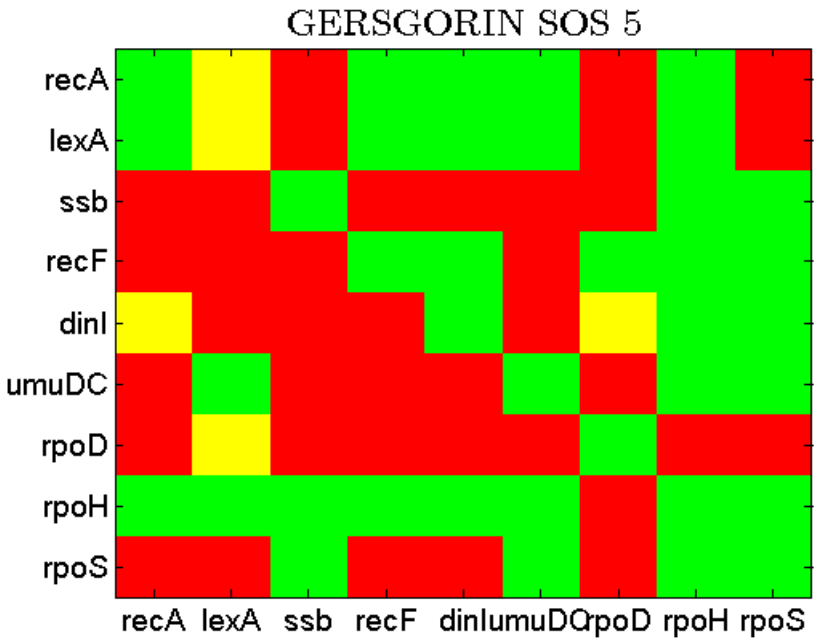



***Figure 4.7.*** *Sign performance evaluation of the SOS network with 5 a priori known interactions, recovered by Gersgorin algorithm with* $t = 0.001$

### Results with 10 known interactions

In this subsection, the priori information corresponds to 10 interactions that are shown in the Table 4.7.

Table 4.8 shows the results of predictions of the three algorithms evaluated in the 71 "unknown" interactions denoted with a ? in Table 4.7. In the Table 4.8 it can be observed that the best performance is also given by the Gersgorin Algorithm when $t \leq 0.01$. In this case none of the predictions falls under the main diagonal in the ROC space in Figure 4.8a. This confirms that the algorithms take advantage of the a priori knowledge which helps the algorithms to reach better identifications of the unknown interactions.

Figure 4.9 shows a sign evaluation of Gersgorin algorithm with $t = 0.005$, which is the best results of the Table 4.8. Compared with Figure 4.7 this figure has much more true recovered interactions which is consistent with measures of performance, $d_{Gers,5,t=0.001} = 0.697$ in front of $d_{Gers,10,t=0.005} = 0.597$.

***Table 4.7.*** *10 Know interactions given to the algorithms.*

| | recA | lexA | ssb | recF | dinI | umuDC | rpoD | rpoH | rpoS |
|---|---|---|---|---|---|---|---|---|---|
| recA | ? | – | ? | ? | ? | ? | ? | ? | ? |
| lexA | ? | – | ? | ? | ? | ? | ? | ? | ? |
| ssb | ? | – | ? | ? | ? | ? | + | ? | ? |
| recF | ? | ? | ? | ? | ? | ? | ? | ? | + |
| dinI | + | ? | ? | ? | ? | ? | + | ? | ? |
| umuDC | ? | ? | ? | ? | ? | ? | + | ? | ? |
| rpoD | + | – | ? | ? | ? | ? | ? | ? | ? |
| rpoH | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| rpoS | ? | ? | ? | ? | ? | ? | ? | ? | ? |

***Table 4.8.*** *Performance metrics of algorithms Unstable, Gersgorin and SDP for the SOS pathway with 10 known edges and different values of the parameter t.*

| | **Unstable** | | | | **Gersgorin** | | | | **SDP** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **t** | **sen** | **spe** | **pre** | **d** | **sen** | **spe** | **pre** | **d** | **sen** | **spe** | **pre** | **d** |
| **0** | 0.475 | 0.548 | 0.463 | 0.693 | 0.425 | 0.839 | 0.773 | 0.597 | 0.475 | 0.548 | 0.463 | 0.693 |
| **0.001** | 0.425 | 0.581 | 0.472 | 0.712 | 0.425 | 0.839 | 0.773 | 0.597 | 0.425 | 0.581 | 0.548 | 0.712 |
| **0.005** | 0.375 | 0.645 | 0.517 | 0.719 | 0.425 | 0.839 | 0.773 | 0.597 | 0.375 | 0.645 | 0.517 | 0.719 |
| **0.01** | 0.3 | 0.839 | 0.706 | 0.718 | 0.325 | 0.806 | 0.684 | 0.702 | 0.35 | 0.71 | 0.609 | 0.712 |
| **0.05** | 0.15 | 0.968 | 0.857 | 0.851 | 0.275 | 0.871 | 0.733 | 0.736 | 0.25 | 0.742 | 0.526 | 0.793 |
| **0.1** | 0.125 | 1 | 1 | 0.875 | 0.225 | 0.968 | 0.9 | 0.776 | 0.225 | 0.903 | 0.75 | 0.781 |
| **0.2** | 0.05 | 1 | 1 | 0.95 | 0.2 | 1 | 1 | 0.8 | 0.2 | 0.935 | 0.8 | 0.803 |
| **0.3** | 0.025 | 1 | 1 | 0.975 | 0.175 | 1 | 1 | 0.825 | 0.225 | 0.935 | 0.75 | 0.778 |
| **0.4** | 0 | 1 | 0 | 1 | 0.175 | 1 | 1 | 0.825 | 0.225 | 0.935 | 0.75 | 0.778 |
| **0.5** | 0 | 1 | 0 | 1 | 0.15 | 1 | 1 | 0.85 | 0.225 | 0.903 | 0.692 | 0.781 |
| **0.6** | 0 | 1 | 0 | 1 | 0.15 | 1 | 1 | 0.85 | 0.225 | 0.903 | 0.692 | 0.781 |
| **0.7** | 0 | 1 | 0 | 1 | 0.15 | 1 | 1 | 0.85 | 0.225 | 0.903 | 0.692 | 0.781 |
| **0.8** | 0 | 1 | 0 | 1 | 0.15 | 1 | 1 | 0.85 | 0.2 | 0.903 | 0.727 | 0.806 |
| **0.9** | 0 | 1 | 0 | 1 | 0.15 | 1 | 1 | 0.85 | 0.225 | 0.935 | 0.818 | 0.778 |
| **1** | 0 | 1 | 0 | 1 | 0.2 | 1 | 1 | 0.8 | 0.225 | 0.935 | 0.818 | 0.778 |

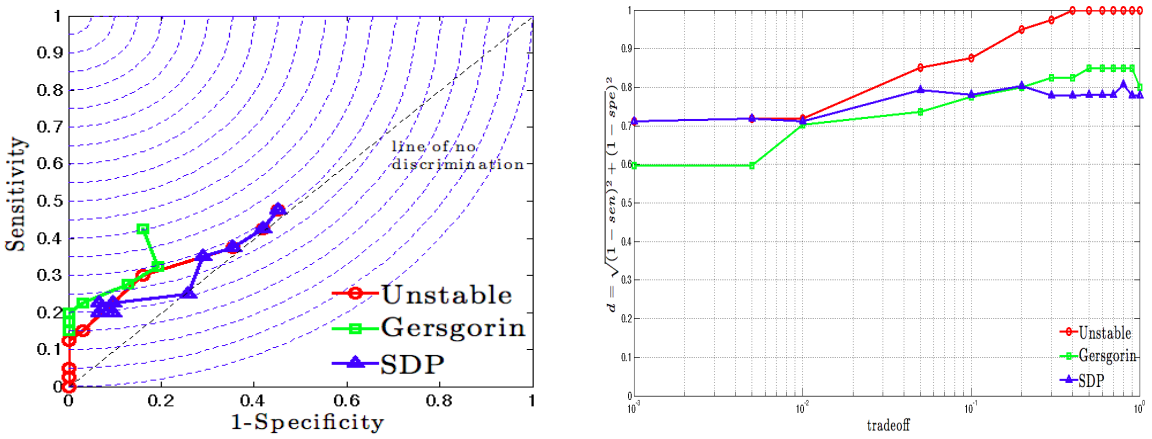



***Figure 4.8.*** *(a) ROC plots of algorithms Unstable (in red), Gersgorin (in green) and SDP (in blue) for the SOS pathway with 10 known edges and different values of the parameter $t$. (b) distance measure $d$ of the algorithms.*

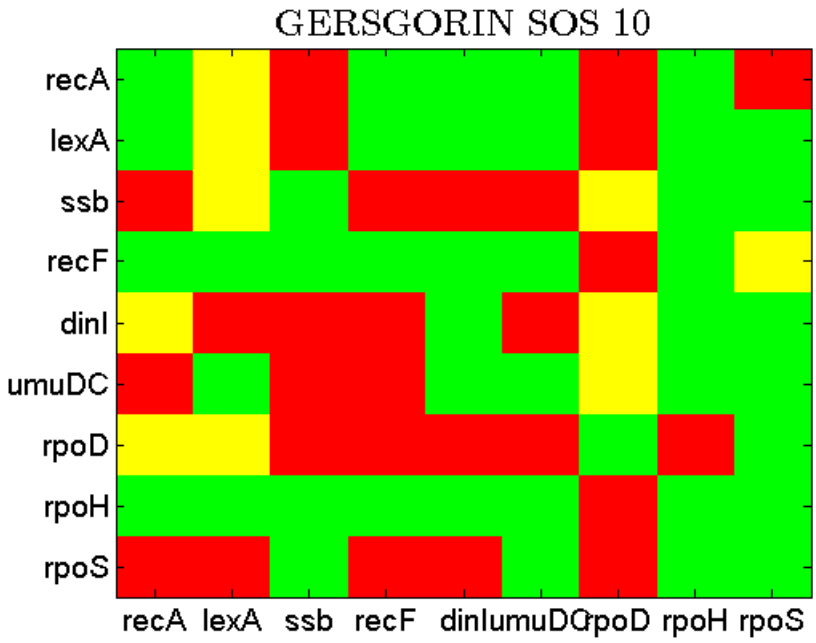



***Figure 4.9.*** *Sign performance evaluation of the SOS network with 10 a priori known interactions, recovered by Gersgorin algorithm with $t = 0.005$*

### Results with 19 known interactions

Finally in this subsection the priori information corresponds to 19 interactions. Table 4.9 shows the results of predictions of the three algorithms evaluated in the 62 "unknown". In the Table it can be observed that the best performance is also given by the Gersgorin Algorithm when $t \leq 0.2$.

***Table 4.9.*** *Performance metrics of algorithms Unstable, Gersgorin and SDP for the SOS pathway with 19 known edges and different values of the parameter $t$.*

| | Unstable | | | | Gersgorin | | | | SDP | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **t** | **sen** | **spe** | **pre** | **d** | **sen** | **spe** | **pre** | **d** | **sen** | **spe** | **pre** | **d** |
| **0** | 0.475 | 0.548 | 0.463 | 0.693 | 0.425 | 0.839 | 0.773 | 0.597 | 0.475 | 0.548 | 0.463 | 0.693 |
| **0.001** | 0.425 | 0.581 | 0.472 | 0.712 | 0.425 | 0.839 | 0.773 | 0.597 | 0.425 | 0.581 | 0.548 | 0.712 |
| **0.005** | 0.375 | 0.645 | 0.517 | 0.719 | 0.425 | 0.839 | 0.773 | 0.597 | 0.375 | 0.645 | 0.517 | 0.719 |
| **0.01** | 0.3 | 0.839 | 0.706 | 0.718 | 0.325 | 0.806 | 0.684 | 0.702 | 0.35 | 0.71 | 0.609 | 0.712 |
| **0.05** | 0.15 | 0.968 | 0.857 | 0.851 | 0.275 | 0.871 | 0.733 | 0.736 | 0.25 | 0.742 | 0.526 | 0.793 |
| **0.1** | 0.125 | 1 | 1 | 0.875 | 0.225 | 0.968 | 0.9 | 0.776 | 0.225 | 0.903 | 0.75 | 0.781 |
| **0.2** | 0.05 | 1 | 1 | 0.95 | 0.2 | 1 | 1 | 0.8 | 0.2 | 0.935 | 0.8 | 0.803 |
| **0.3** | 0.025 | 1 | 1 | 0.975 | 0.175 | 1 | 1 | 0.825 | 0.225 | 0.935 | 0.75 | 0.778 |
| **0.4** | 0 | 1 | 0 | 1 | 0.175 | 1 | 1 | 0.825 | 0.225 | 0.935 | 0.75 | 0.778 |
| **0.5** | 0 | 1 | 0 | 1 | 0.15 | 1 | 1 | 0.85 | 0.225 | 0.903 | 0.692 | 0.781 |
| **0.6** | 0 | 1 | 0 | 1 | 0.15 | 1 | 1 | 0.85 | 0.225 | 0.903 | 0.692 | 0.781 |
| **0.7** | 0 | 1 | 0 | 1 | 0.15 | 1 | 1 | 0.85 | 0.225 | 0.903 | 0.692 | 0.781 |
| **0.8** | 0 | 1 | 0 | 1 | 0.15 | 1 | 1 | 0.85 | 0.2 | 0.903 | 0.727 | 0.806 |
| **0.9** | 0 | 1 | 0 | 1 | 0.15 | 1 | 1 | 0.85 | 0.225 | 0.935 | 0.818 | 0.778 |
| **1** | 0 | 1 | 0 | 1 | 0.2 | 1 | 1 | 0.8 | 0.225 | 0.935 | 0.818 | 0.778 |

Comparing Table 4.9 with Table 4.8, we can see that introducing 9 new known interactions does not help to improve the performance identification, $d_{Gers,10,t=0.005} = 0.597$ in front of $d_{Gers,19,t=0.005} = 0.597$. This effect could be caused because there are less unknown interactions to recover. Some of the results of the Unstable algorithm and the SDP fall under the main diagonal in the ROC space at Figure 4.10. This results are obtained with high values of $t$, giving therefore more weight to the model.

We have presented the results and conclusions for the different algorithms, taking some insight in the characteristics of each one. Even though, this results should be confirmed with other dataset. This step is done in the next subsection.

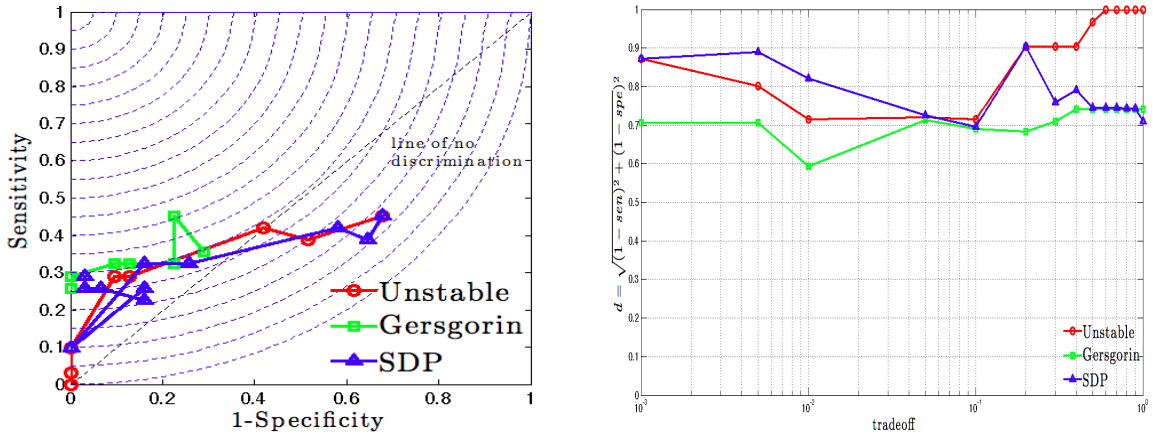



***Figure 4.10.*** *(a) ROC plots of algorithms Unstable (in red), Gersgorin (in green) and SDP (in blue) for the SOS pathway with 19 known edges and different values of the parameter t. (b) distance measure d of the algorithms.*

### 4.2.2 Synthetic data

We performed a comparison using 'fake' gene expression data generated by a computer model of gene regulation ('in silico' data). In silico data, which is an expression used in biology to mean "performed on computer or via computer simulation", enable one to check the performance of algorithms against a perfectly known ground truth (simulated networks in the computer model). The need of simulated data arises from imperfect knowledge of real networks in cells, from the lack of suitable gene expression data set and of control on the noise levels.

The artificial data has to be independent of the reverse engineering algorithm to avoid a bias in the results. Therefore, the data selected to perform the analysis consist in 20 in-silico datasets used in [15], these datasets are generated with ODEs from stable and well-conditioned networks.

In order to evaluate the performance of the algorithms, we have used the data of local interventions (a different single gene in the network is perturbed in each experiment). There are 20 different networks with 10 genes. In this case, the average of connectivity of the different networks is up to 27%, which is an expected value in real biological networks.

We have performed our analysis with different noise models:

- Original: Additive Gaussian Noise, with zero mean and standard deviation equal to $0.1$.
- Noisy: The original data is additionally corrupted with Gaussian noise with zero mean and variance equal to $0.25$.

### Original data

The mean results of the performance metrics obtained with the original in-silico datasets are presented in the following subsection. For each of the results, we calculated average values across the 20 networks. In the tables we present the mean values and in the figures the standard deviation values are also shown.

### Z-score and |Z-score|

Results obtained with Z-score and |Z-score| algorithms are presented respectively in Table 4.10 and Table 4.11. We observe that for this data set, both algorithms achieve better performances that the ones obtained with the SOS pathway, for some thresholds even the original Z-score algorithm behaves reasonably well. The results are also shown in Figure 4.11, it can be observed that the best result is obtained as in the case of SOS pathway with the |Z-score| algorithm. This is to be expected, since it only cares about the topology of the network, with this algorithm we can obtain a quite reasonable distance of $d = 0.3694$ taking in account the simplicity of the algorithm and its very low computational complexity.

***Table 4.10.*** *Performance results of Z-score algorithm for the original in-silico dataset and different values of the threshold* $\gamma$

| $\gamma$ | $sen$ | $spe$ | $d$ |
|---|---|---|---|
| 0.49164 | 0.51303 | 0.0056251 | 1.1087 |
| 2.7756 | 0.32884 | 0.84056 | 0.69872 |
| 5.0596 | 0.11845 | 0.97826 | 0.8827 |
| 7.3436 | 0.050712 | 0.99301 | 0.94941 |
| 9.6276 | 0.024582 | 0.99862 | 0.97543 |
| 11.9116 | 0.012201 | 0.99932 | 0.9878 |
| 14.1956 | 0.0052427 | 1 | 0.99476 |
| 16.4796 | 0 | 1 | 1 |
| 18.7636 | 0 | 1 | 1 |
| 21.0476 | 0 | 1 | 1 |

***Table 4.11.*** *Performance results of |Z-score| algorithm for the original in-silico dataset and different values of the threshold $\gamma$*

| $\gamma$ | *sen* | *spe* | *d* |
|---|---|---|---|
| 0.49164 | 0.97825 | 0.0056251 | 0.99477 |
| 2.7756 | 0.68888 | 0.84056 | 0.3694 |
| 5.0596 | 0.46588 | 0.97826 | 0.53621 |
| 7.3436 | 0.36703 | 0.99301 | 0.63317 |
| 9.6276 | 0.28858 | 0.99862 | 0.71144 |
| 11.9116 | 0.21084 | 0.99932 | 0.78916 |
| 14.1956 | 0.1421 | 1 | 0.8579 |
| 16.4796 | 0.091155 | 1 | 0.90885 |
| 18.7636 | 0.057091 | 1 | 0.94291 |
| 21.0476 | 0 | 1 | 1 |

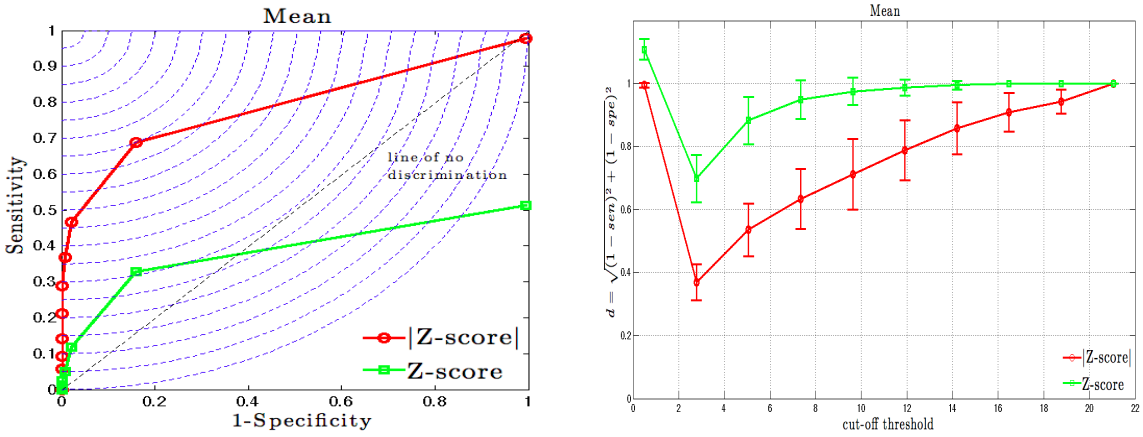



***Figure 4.11.*** *(a) ROC plots of z-score algorithms for the in-silico dataset from [15], and different values of the parameter $\gamma$. (b) Distance measure d of the algorithms, in the curves are shown the mean and standard deviation.*

### Zavlanos algorithms

The three different Zavlanos algorithms take advantage of the a priori knowledge about the network in the form of a partial sign pattern between all pairwise genes in the networks (matrix $S$ of equation 3.13). In order to evaluate the performance of the algorithms not all the a priori knowledge is used and the amount of the considered unknown interactions is set to 10 of the 100 possible interactions.

In the Table 4.12 the performance results of the different Zavlanos algorithms obtained with the in-silico dataset are collected. Similarly to the results presented for the real network, these methods obtain better results than the z-score method.

In Figure 4.12a it can be noticed that in contrast with z-score none of the methods and none of the different tradeoff values $t$ return a network that performs worse than a randomly generated network. In comparison with the SOS pathway, in Figure 4.12b it can be observed that the best results are recovered with SDP algorithm. This observation is an indication that stability is important, since the best results are the ones obtained with SDP in this dataset and with Gersgorin in the SOS pathway. Since the mean connectivity is more sparse than for the SOS pathway and the SNR is much higher, the best tradeoff values defined by $t$ are higher giving therefore less importance to the error fit and more to the network model.

***Table 4.12.*** *Mean of performance metrics of Zavlanos algorithms for the original in-silico dataset from [15] and different values of the parameter $t$.*

| | **Unstable** | | | **Gersgorin** | | | **SDP** | | |
|---|---|---|---|---|---|---|---|---|---|
| **t** | **sen** | **spe** | **d** | **sen** | **spe** | **d** | **sen** | **spe** | **d** |
| **0** | 0.998 | 0.60697 | 0.3932 | 0.9345 | 0.64912 | 0.36578 | 0.998 | 0.60697 | 0.3932 |
| **0.001** | 0.998 | 0.60776 | 0.39241 | 0.9345 | 0.64912 | 0.36578 | 0.998 | 0.60776 | 0.39241 |
| **0.005** | 0.998 | 0.6125 | 0.38767 | 0.9345 | 0.65073 | 0.36417 | 0.998 | 0.6125 | 0.38767 |
| **0.01** | 0.998 | 0.62422 | 0.37595 | 0.9345 | 0.65234 | 0.36256 | 0.998 | 0.62422 | 0.37595 |
| **0.05** | 0.998 | 0.72282 | 0.27746 | 0.93279 | 0.67112 | 0.34416 | 0.998 | 0.72282 | 0.27746 |
| **0.1** | 0.99244 | 0.7861 | 0.21683 | 0.94087 | 0.66762 | 0.34324 | 0.99244 | 0.7861 | 0.21683 |
| **0.3** | 0.97724 | 0.87074 | 0.14017 | 0.93478 | 0.68887 | 0.32937 | 0.97902 | 0.87074 | 0.13964 |
| **0.5** | 0.90511 | 0.78092 | 0.25505 | 0.91933 | 0.72616 | 0.30153 | 0.91686 | 0.78092 | 0.25117 |
| **0.8** | 0.6526 | 0.8474 | 0.39675 | 0.74206 | 0.82327 | 0.32933 | 0.76433 | 0.84967 | 0.2948 |
| **1** | 0 | 1 | 1 | 0.36644 | 1 | 0.63356 | 0.30371 | 0.99926 | 0.69629 |

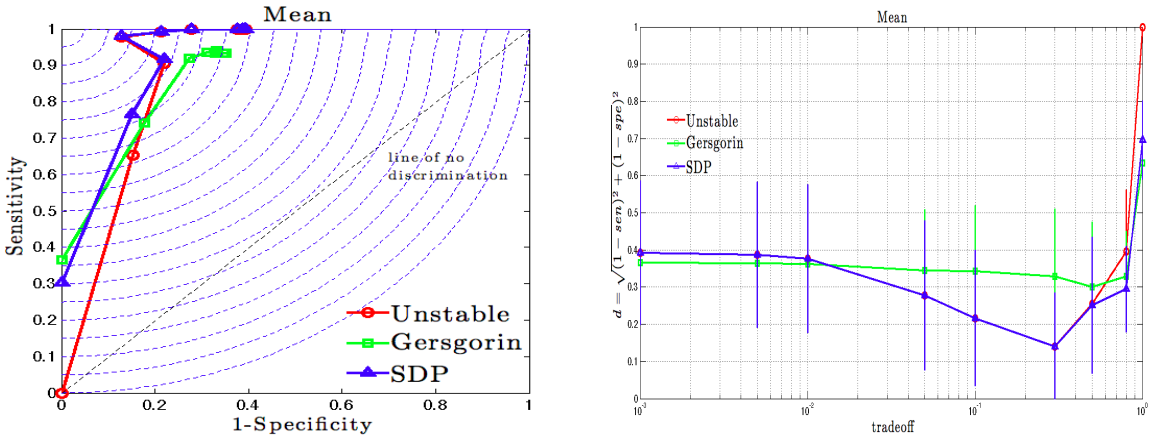



***Figure 4.12.*** *(a) ROC plots of algorithms Unstable (in red), Gersgorin (in green) and SDP (in blue) for the original in-silico dataset from [15], and different values of the parameter t. (b) Distance measure d of the algorithms, in the curves are shown the mean and standard deviation.*

### Noisy data

Here we present the mean results of the performance metrics obtained with the in-silico datasets corrupted with additional Gaussian noise with zero mean and variance equal to $0.25$. As in the previous subsection, for each of the results, we calculate an average across the 20 networks. In the tables we present the mean values and in the figures the standard deviation values are also shown.

### Z-score and |Z-score|

The results obtained with Z-score and |Z-score| are presented respectively in Table 4.13 and Table 4.14. The results are also shown in Figure 4.13, in this case the performance of the algorithms are worse than the ones obtained with the original datasets as could be expected. In the case of |Z-score| the algorithm obtains a better result than random guessing for a large values of threshold. However, the original Z-score algorithm is not able to recover a reasonably good network and only with very small threshold it behaves like a random guess, this results confirms our conclusions extracted in the SOS pathway section 4.2.1.

***Table 4.13.*** *Performance results of Z-score algorithm for the original in-silico dataset and different values of the threshold $\gamma$*

| $\gamma$ | *sen* | *spe* | *d* |
|---|---|---|---|
| 0.063429 | 0.46436 | 0.01182 | 1.1256 |
| 1.1433 | 0.33788 | 0.44745 | 0.86644 |
| 2.2231 | 0.2245 | 0.70634 | 0.83468 |
| 3.3029 | 0.12468 | 0.83864 | 0.89336 |
| 4.3828 | 0.075003 | 0.91198 | 0.9306 |
| 5.4626 | 0.040841 | 0.94904 | 0.96109 |
| 6.5424 | 0.024784 | 0.97694 | 0.97567 |
| 7.6222 | 0.014123 | 0.9881 | 0.98602 |
| 8.7021 | 0.0072222 | 0.99366 | 0.99283 |
| 9.7819 | 0 | 1 | 1 |

***Table 4.14.*** *Performance results of |Z-score| algorithm for the in-silico dataset corrupted by noise and different values of the threshold $\gamma$*

| $\gamma$ | *sen* | *spe* | *d* |
|---|---|---|---|
| 0.063429 | 0.99476 | 0.01182 | 0.98827 |
| 1.1433 | 0.79187 | 0.44745 | 0.59787 |
| 2.2231 | 0.61045 | 0.70634 | 0.49758 |
| 3.3029 | 0.47535 | 0.83864 | 0.55478 |
| 4.3828 | 0.3436 | 0.91198 | 0.66445 |
| 5.4626 | 0.23486 | 0.94904 | 0.76758 |
| 6.5424 | 0.14264 | 0.97694 | 0.85787 |
| 7.6222 | 0.0869 | 0.9881 | 0.91325 |
| 8.7021 | 0.047894 | 0.99366 | 0.95216 |
| 9.7819 | 0 | 1 | 1 |

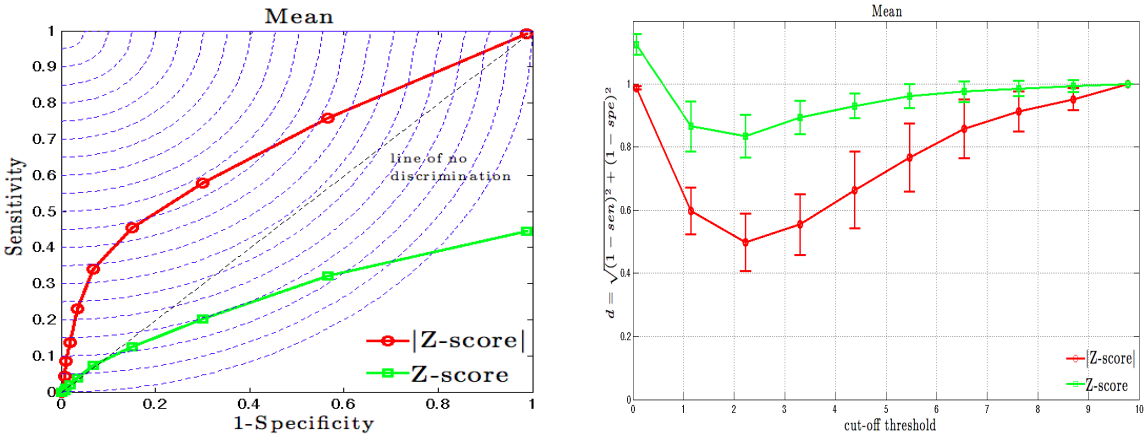



***Figure 4.13.*** *(a) ROC plots of z-score algorithms for the in-silico dataset from [15] corrupted with additional Gaussian noise, and different values of the parameter t. (b) Distance measure d of the algorithms, in the curves are shown the mean and standard deviation.*

### Zavlanos algorithms

In the Table 4.15 the performance results of the different Zavlanos algorithms obtained with the in-silico dataset corrupted with additional Gaussian noise are collected. Similarly to the results presented before, these methods obtain better results than the z-score method.

In Figure 4.14a it can be noticed that some of the thresholds return networks estimates that are worse than a randomly selected network. In this case of noisy data, the best algorithm is the Gersgorin as in the SOS pathway database, as it can be observed in Figure 4.14b.

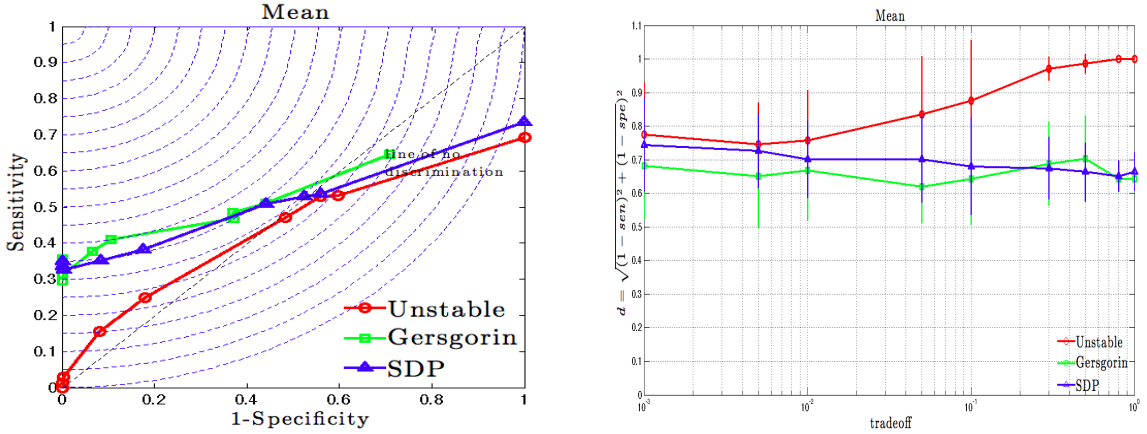



***Figure 4.14.*** *(a) ROC plots of algorithms Unstable (in red), Gersgorin (in green) and SDP (in blue) for the in-silico dataset from [15] corrupted with additional Gaussian noise, and different values of the parameter t. (b) Distance measure d of the algorithms, in the curves are shown the mean and standard deviation.*

***Table 4.15.*** *Mean of performance metrics of Zavlanos algorithms for the in-silico dataset from [15] corrupted with additional noise, and different values of the parameter t.*

| | Unstable | | | Gersgorin | | | SDP | | |
|---|---|---|---|---|---|---|---|---|---|
| **t** | **sen** | **spe** | **d** | **sen** | **spe** | **d** | **sen** | **spe** | **d** |
| **0** | 0.69232 | 0 | 1.055 | 0.64717 | 0.29055 | 0.816 | 0.73434 | 0.0015165 | 1.0395 |
| **0.001** | 0.53097 | 0.40201 | 0.77474 | 0.51094 | 0.55986 | 0.68234 | 0.53638 | 0.44122 | 0.74448 |
| **0.005** | 0.52917 | 0.44107 | 0.74567 | 0.48423 | 0.63042 | 0.65034 | 0.52843 | 0.47697 | 0.72715 |
| **0.01** | 0.471 | 0.5176 | 0.75739 | 0.46745 | 0.62927 | 0.66803 | 0.50754 | 0.55809 | 0.70152 |
| **0.05** | 0.24776 | 0.8196 | 0.83603 | 0.41035 | 0.89353 | 0.62013 | 0.38232 | 0.82428 | 0.70136 |
| **0.1** | 0.15623 | 0.91902 | 0.87506 | 0.37767 | 0.93483 | 0.64397 | 0.35075 | 0.91675 | 0.68061 |
| **0.3** | 0.02836 | 0.99768 | 0.97166 | 0.31203 | 0.99847 | 0.68798 | 0.32539 | 0.99769 | 0.67463 |
| **0.5** | 0.013894 | 0.99928 | 0.98611 | 0.29665 | 1 | 0.70335 | 0.33602 | 0.99848 | 0.664 |
| **0.8** | 0 | 1 | 1 | 0.35676 | 1 | 0.64324 | 0.34856 | 1 | 0.65144 |
| **1** | 0 | 1 | 1 | 0.35676 | 1 | 0.64324 | 0.33502 | 1 | 0.66498 |

## 4.3 Conclusions of Unstable, Gersgorin and SDP algorithms

In the previous sections we have evaluated the minimal model identification that best explains genetic perturbation data obtained at the network's equilibrium state. We have tested the performance and sensitivity of the different algorithms to parameter selection, for various amount of the a priori knowledge. It could be observed that the Gersgorin algorithm has worked better than the Unstable and SDP algorithms in the SOS pathway dataset.

On the other hand, in the in-silico dataset the best results have been obtained with the SDP algorithm in the original dataset and with Gersgorin in the case of noisy data. This observation is an indication that stability is important, not only for consistency with the problem assumptions, but also for identification performance. The best identifications correspond to values of the parameter t when $t \in \{0, 0.1\}$, which is a relative small weight to the model (equation 3.12). We have also observed that the amount of a priori information allows the different algorithms to reach better predictions of the unknown interactions.

# Chapter 5

# Algorithms prediction evaluation and identification boosting performance method

This chapter presents the contribution of this thesis to the state of the art technique analyzed in the previous chapter. The contribution is divided in two fields. First, we present a prediction analysis of the Zavlanos methods of reverse engineering of gene networks. Second, we introduce a framework to boost the performance of the presented reverse engineering methods, based in ensemble decision methods explained in [51].

## 5.1 Prediction analysis

In order to estimate how accurately the inferred network would perform in practice as a prediction model, we have used a leave-one-out cross-validation approach with the SOS dataset used in section 4.2.1.

The protocol is the following, it uses a single intervention experiment from the original sample as the validation data, and the remaining experiments as the training data in order to infere the network. Then, this network is used to predict the outcome of the hidden experiment and then the predictions are compared in terms of sign. For example, if in the experiment the gene $i$ increases its level of expression, and if the the network predicts that situation, this is considered as a correct prediction. This is repeated such that each experiment in the database is used once as the validation data, in the case of SOS pathway this is done up to nine times, the process is illustrated in Figure 5.1. Leave-one-out cross-validation is computationally expensive

because it requires many repetitions of the training step.

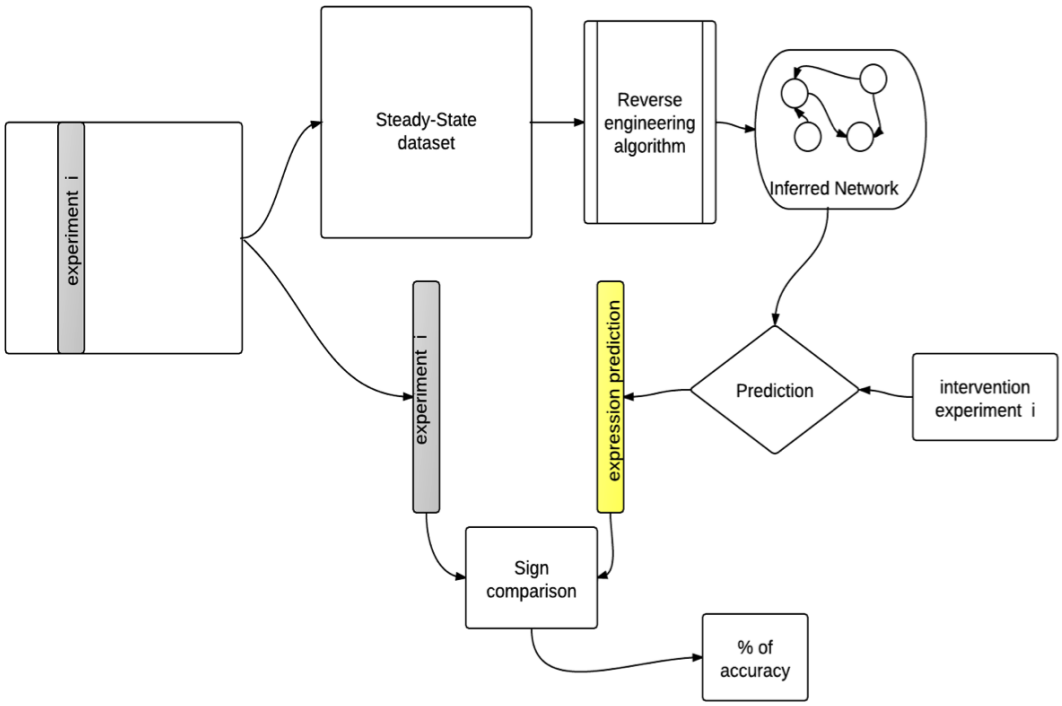



***Figure 5.1.*** *Validation strategies for network prediction methods.*

This analysis has been performed for the Zavlanos algorithms for different tradeoff values between sparse model and error fit. In the following figures, the mean distance d to the perfect classification (equation 4.2) obtained in the cross-validation is represented in a blue line and the percentage of mean prediction error in a red line. Figures 5.2, 5.3 and 5.4 show respectively the results for Unstable, Gersgorin and SDP algorithms. At the end, this protocol implies to identify nine networks for each one of three algorithms and each of fifteen different values of $t$, so the total number of simulations is 405.

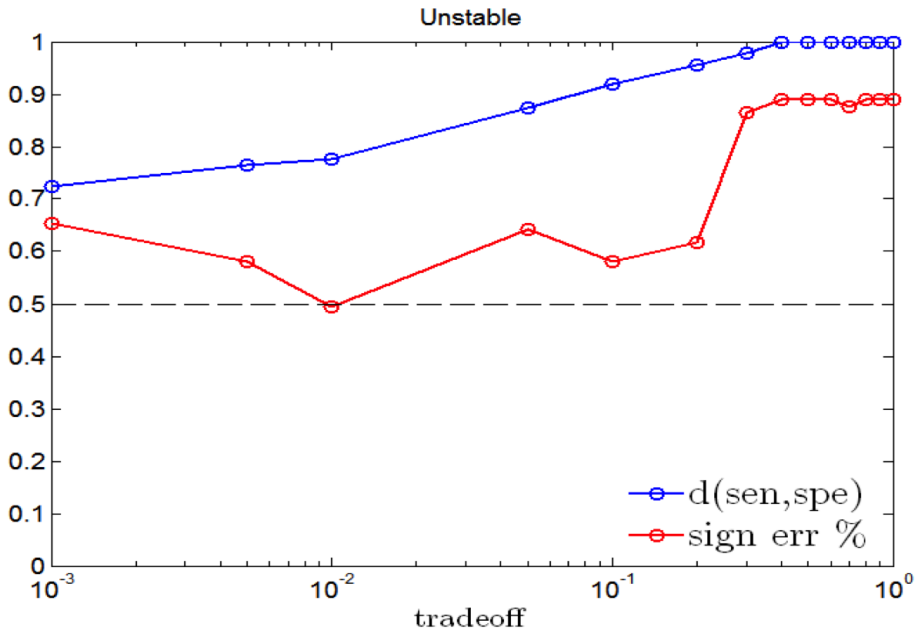



***Figure 5.2.*** *Prediction analysis for Unstable algorithm, in the SOS pathway.*

In Figure 5.2 it can be observed that only one value achieves a 50% of prediction accuracy, moreover the best value does not coincide with the best performance in terms of distance to best classification, which is an indication that a small variation in the network implies a big change in the prediction of an intervention.

In the case of Gersgorin algorithm, its performance is shown in Figure 5.3. It can be noticed that in this case for values of $t \leq 0.1$ the recovered networks are able to predict the experiment with an accuracy of the 60%.

In the case of SDP algorithm, its prediction ability is more moderate than the Gersgorin algorithm but better than Unstable. Its performance is shown in Figure 5.4.

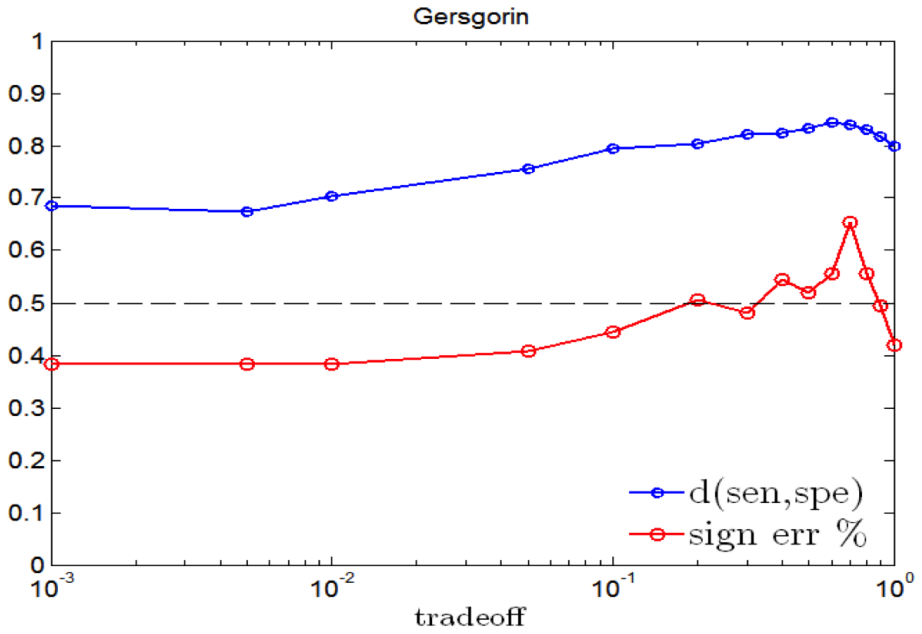



***Figure 5.3.*** *Prediction analysis for Gersgorin algorithm, in the SOS pathway.*

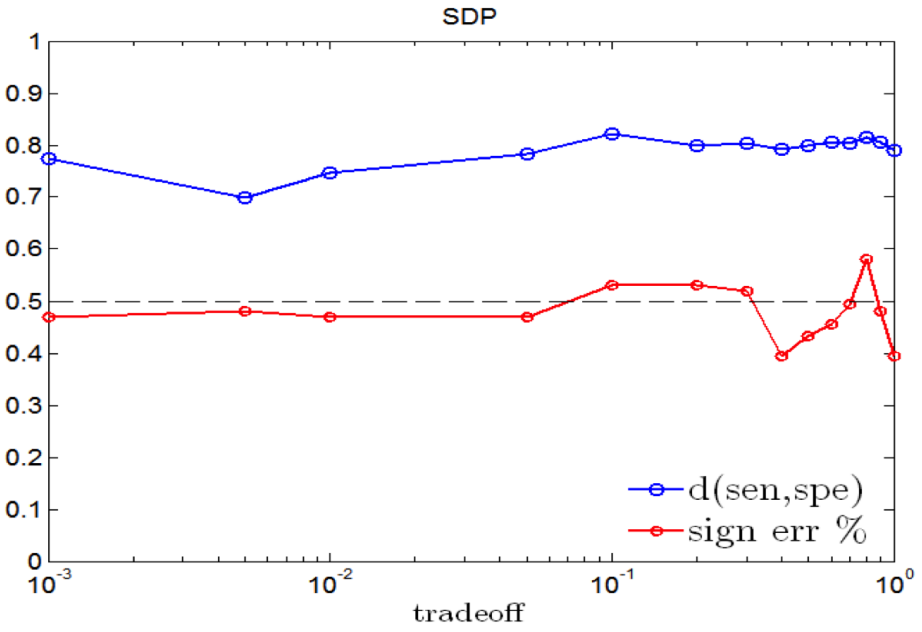



***Figure 5.4.*** *Prediction analysis for SDP algorithm, in the SOS pathway.*

### 5.1.1 Model prediction conclusions

Here we have tackle the strength of the inferred gene network relating it to its ability to explain the different experiments and the behavior of the genes. We have noticed that the stability constrain has also been crucial to achieve better predictions. As in the case of topology inference, in the case of the SOS pathway the best algorithm is the one based on the theorems of Gersgorin.

## 5.2 Voting gene networks

As explained before, the Zavlanos algorithms for inferring biological networks are based on the fitting of a mathematical model to a dataset of experimentally observed activity levels of the network components while being the output consistency with some prior knowledge. We have observed that there are many different networks that could be recovered with different values of tradeoff that are consistent with the data.

Here we present a preliminary idea to obtain a better network based on the ideas presented by [52]. Instead of identifying a unique "best" network, we want to integrate the different networks and make a global decision based on the different networks taking advantage of the different characteristics of the algorithms. This idea is illustrated in the Figure 5.5, which is adapted from [52].

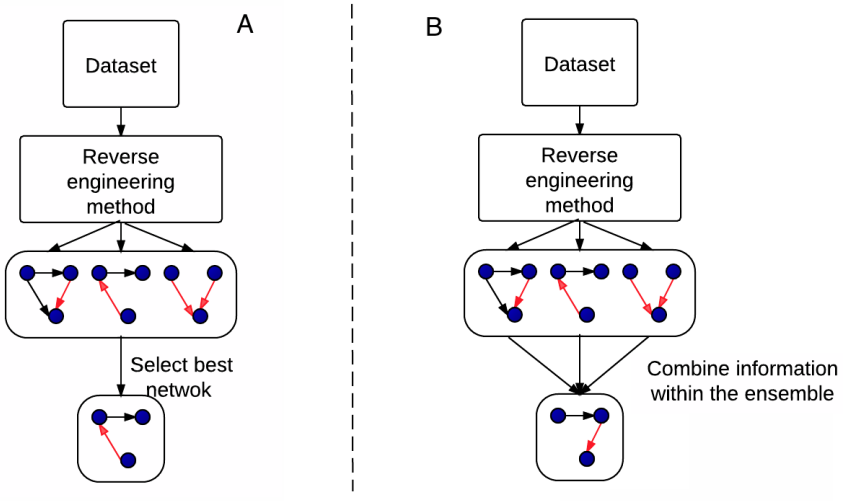



***Figure 5.5.*** *(A) Identification of the single "best" network. (B) Integration of the information from different networks in order to make one better prediction, this method could also be used to estimate the reliability of these predictions. Figure adapted from [52].*

### 5.2.1 Ensemble Methods

The classic example of an ensemble-based system in decision making is the popular game show "Who Wants to Be a Millionaire?", which is just one of many examples in which ensembles of diverse individuals outperform a single expert on average. In general terms, the aggregation of information in groups, result in decisions that are often better than the one that could have been made by any single member of the group [51].

Consider an ensemble of inferred networks obtained by a gene network reverse-engineering method from a dataset of gene expression measurements. Each of these networks is a hypothesis of the true network structure, giving a prediction on the presence or absence of a regulatory link for every pair of genes. Now, assume that the prediction of links is correct with probability $P > 0.5$ (better than random guessing) and that the errors between the different networks of the ensemble are uncorrelated. In this case, the prediction obtained from the ensemble by voting (see next section) is on average more accurate than any of the individual networks of the ensemble [53].

### 5.2.2 Signed Voting on the Network Structure

The first simple voting scheme, which we call signed voting, can predict the signed regulatory links from an ensemble of inferred networks.

If we represent the network structures by a matrix $A$, where $a_{ij} = 1$ if the link is excitatory, $a_{ij} = -1$ if the link is inhibitory, and $a_{ij} = 0$ if the link is absent. Then for each value of threshold we have an ensemble of 3 networks corresponding with the different algorithms (Unstable, Gersgorin, SDP). The structure of one of this network is defined by the matrix $A^k$, the signed vote for each link in the networks is defined by the following equation:

$$v_{ij} = \frac{\sum_{k=1}^{3} a_{ij}^k}{3} \tag{5.1}$$

We need to determine when a connection is set to zero. To do this, a link is set to zero if the absolute value of the signed vote is smaller than some threshold $|v_{ij}| < c$. The smaller $c$, the more uncertain links are included in the final network prediction. In this work we have set $c = 0.25$ because with this value we have obtained the best results.

This voting method treats the excitatory and inhibitory links as opposite.

### 5.2.3 Mode Voting on the Network Structure

Another straightforward approach is majority voting. Every network of the ensemble votes on the classification of a given link as excitatory, inhibitory, or absent. The type of link is defined by the majority of the votes. When there are multiple values occurring equally frequently, the value is set randomly between them. But since we perform a mode between three algorithms this situation only happens if the three doesn't agree.

This voting method treats the three possible types of a link (excitatory, inhibitory, and zero) as equal.

### 5.2.4 Experimental Results

We use the same benchmark protocol and networks as in section 4.2.2 as an example for demonstrating the potential of ensemble approaches in gene-network reverse engineering.

The resulting performance of the different methods to infer gene networks are shown in this subsection. In this case the information is only shown in a graphic form as the average across the 20 networks, showing the mean and the standard deviation values. The legend of the figures is the following: in red the results of Unstable algorithm, in green the results obtained by Gersgorin, in blue by the SDP, in magenta and labeled as "Signed" the results obtained with Signed Voting on the Network Structure, and finally in black and named as "Mode" the results obtained by the Mode Voting on the Network Structure method.

In Figure 5.6 presents the ROC plot and distance to the perfect reconstuction point for the original in-silico datasets. It can be observed that none of the ensemble methods is able to improve the results. In this case the mean voting follows the Gersgoring performance and the mode voting follows the SDP performance. Since the results obtained by the original Zavlanos algorithms are very good we think that there is no real margin to improve with this very simple voting rules.

In Figure 5.7 shows the ROC plot and distance to the perfect reconstuction point for the in-silico datasets with additional noise. In this case the mode voting approach is able to improve the identification performance of the original methods for values of $t \leq 0.1$. We think that this could be caused because with this method of voting the excitatory and inhibitory votes cancel each other out, whereas votes for the absence of a link are neutral, this could be a drawback. For the same margin of $t$ the mean voting scheme goes worse than the original methods, but with $t > 0.1$ all the stable and voting methods converge to the same identification performance.

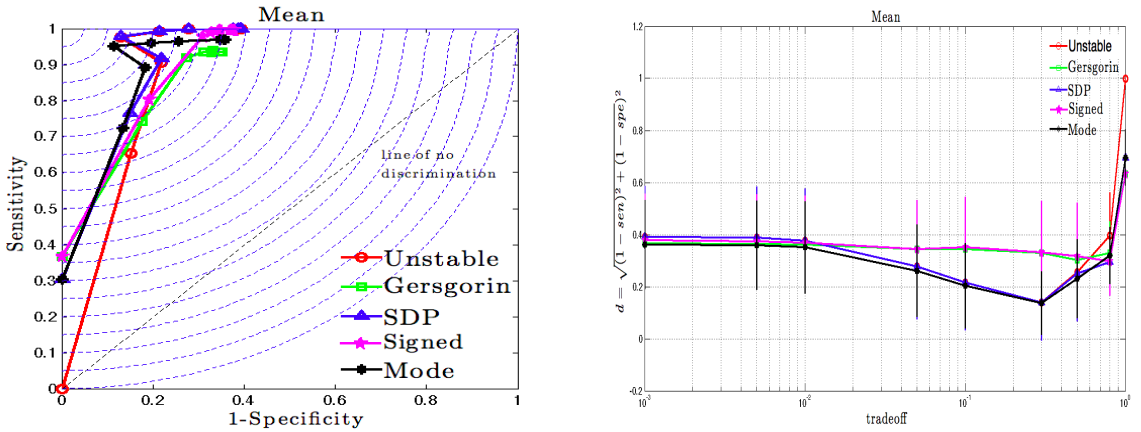



***Figure 5.6.*** *(a) ROC plots of algorithms for the original in-silico dataset and different values of the parameter t. (b) distance measure d of the algorithms, in the curves are shown the mean and standard deviation.*

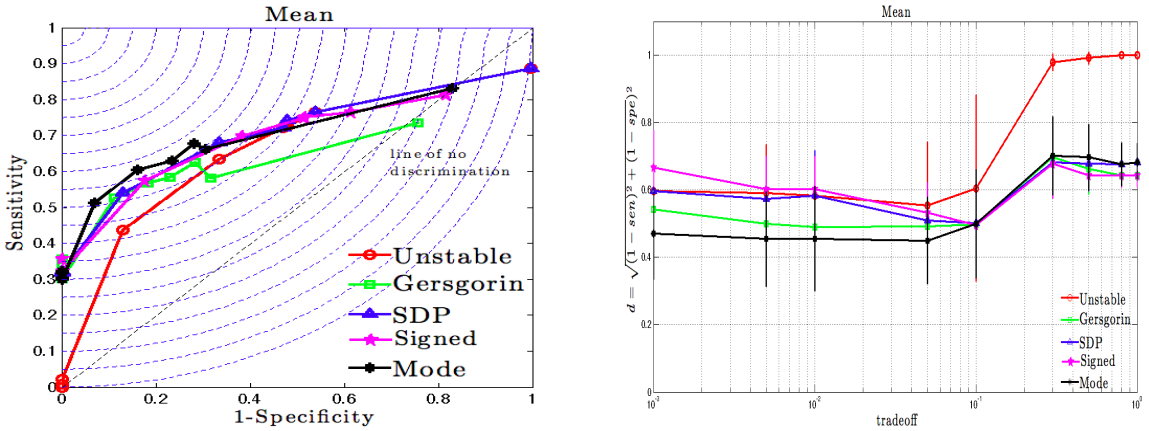



***Figure 5.7.*** *(a) ROC plots of algorithmsf or the in-silico dataset corrupted with additional noise and different values of the parameter t. (b) distance measure d of the algorithms.*

### 5.2.5 Conclusions of ensemble voting

The ensemble voting boosts performance compared to individual members of the ensemble if the prediction errors are uncorrelated. It seems very unlikely that this statement holds in an ensemble of networks that are inferred from the same dataset (in spite that they are inferred with different methods). This could be the reason because we haven't see a dramatically improvement. Yet, we have found some promising results that show an improvement on the accuracy of predictions.

In the future othermore powerful voting techniques should be designed, weighting the votes linearly or with a non-linear technique.

# Chapter 6

# Conclusions and future work

This final chapter reviews the project achievements and the obtained conclusions. Also, some future work lines are presented based on the open issues detected during the project development.

## 6.1 Conclusions

In this work, the characteristics of the microarray datasets have been presented and also, their utility for measuring the gene expression for large scale biological networks. We have explained that they provide valuable data that can be used to identify gene interactions in large genetic networks. The discovery of these networks are also important in drug discovery, where a understanding of regulatory networks is crucial for identifying the targeted pathways [54].

We have reviewed different approaches to recover gene networks, in chapter 3. Pointing out the advantages and drawback of different solutions to tackle the problem. After this revision we have chosen two different approaches, one which model the networks as differential equations and aims at obtaining a minimal model that explains the interventions, the Zavlanos algorithms, and another due to its extreme simplicity and good performance, the z-score algorithm.

A protocol and metrics have been defined in order to evaluate the identification performance of these different alternatives, which are tested individually and their results are presented in chapter 4. We have made an extensive analysis of the chosen methods, testing their performance in the task of identifying the networks that explains the datasets. Two different datasets have been used in order to perform this task. The first one is a real dataset consisting in steady-state experiments with knowledge of the perturbed gene in each experiment, that comes from a nine-gene network. The second one is a fake gene expression dataset that

simulates steady-state experiments of 20 different well-conditioned networks. The best results have been obtained with the SDP algorithm in the original dataset and with Gersgorin in the case of noisy data and the real dataset. As a result, the Gersgorin algorithm is the preferred option in practice. Moreover, with both datasets we have found that stability constraint in the identified network is important, first for consistency with the problem assumptions [23, 24], and also for obtaining a better identification performance. We have also observed that the amount of a priori information allows the different algorithms to reach better predictions of the unknown interactions.

After reviewed the ability to identify the network topology, a different method has been presented in order to evaluate the networks in chapter 5. It is based in the ability of the inferred network to predict the behavior of the system following of a given perturbation, that is, we want to know whether the gene network models can be used to predict the response of a network to an external perturbation. With this type of analysis of the algorithm we have also demonstrated the usefulness of the stability constraint, although we have reached quite moderate predictions due to the simplicity of the model. Also in chapter 5 a new approach to boost the identification performance has been presented. It is based on an ensemble decision paradigm. It is a preliminary idea but even though, we have found some promising results that demonstrate the potential of the approach.

## 6.2 Future work lines

In this section, some future work lines are presented as possible continuation paths from the current stage.

As discussed in the chapter 5, ensemble voting boosts performance compared to a individual members of the ensemble in a noisy environment. As a result, we think that there is margin to improve the performances, therefore another work line is to study other voting techniques that based on non-linear combination rules boost the performance in the non-noisy datasets as in section 4.2.2.

Also, an additional work line is to combine the results from different approaches. We have seen that the |z-score| recovers the existence of a link with very good precision, therefore the output of this algorithm could be used in form of linear constraint as in equation 3.13 or any other way.

Comparing the results presented in section 4.2.1 and 4.2.2, there is a dramatically improvement in the recognition performance. We think that this is caused by the simplicity of the regulation

model based on lineal ODEs. Therefore, a possible work line is to change the model and use for example a standard thermodynamic approach allowing for both independent ('additive') and synergistic ('multiplicative') regulatory interactions. For each gene $i$ of a network, the rate of change of mRNA concentration $F_i^{RNA}$ and the rate of change of protein concentration $F_i^{Prot}$ are described by :

$$\begin{aligned} F_i^{RNA}(\mathbf{x},\mathbf{y}) =& \frac{\mathrm{d}x_i}{\mathrm{d}t} = m_i f_i(\mathbf{y}) - \lambda_i^{RNA} x_i \\ F_i^{Prot}(\mathbf{x},\mathbf{y}) =& \frac{\mathrm{d}y_i}{\mathrm{d}t} = r_i x_i - \lambda_i^{Prot} y_i \end{aligned} \tag{6.1}$$

Where $m_i$ is the maximum transcription rate, $r_i$ the translation rate, $\lambda_i^{RNA}$ and $\lambda_i^{Prot}$ are the mRNA and protein degradation rates and $\mathbf{x}$ and $\mathbf{y}$ are vectors containing all mRNA and protein concentration levels, respectively. $f_i(\cdot)$ is the activation function of gene $i$, which computes the relative activation of the gene, which is between 0 (the gene is shut off) and 1 (the gene is maximally activated), given the protein or TF concentrations $\mathbf{y}$. A more detailed description of the activation function can be found in [11, 55].

However, there is a caveat, since the complexity of the model has increased it is more difficult to fit the dynamical model and moreover if we only have acces to steady-state datasets. A novel approach, named *linkage logic*, presented in [56] allows to restrict possible steady states of a given complex network system from the knowledge of regulatory linkages alone, allowing to analyze the complex dynamics of network systems. Therefore, we think that this method or similiar ones could be included in our algorithm in order to improve the performance.

Additionally, the big challenge is to work with time-series experiments, this would open whole new field of action. There are different approaches that are not explained in this work because they are only focused to this kind of data. This is the case of the TSNI (Time Series Network Identification) algorithm [57] which identifies the gene network ($a_{ij}$) as well as the direct targets of the perturbations ($b_i$). Many other works deal with this kind of data and the issues to fit a standard thermodynamic model to the data (for example [58]). As it is claimed in [59, 60] these type of data contain different and complementary information (with respect to steady-state datasets) about the network. That could lead to a different integration model that takes advantage of the different strengths of each data and reverse engineering approaches.

# Appendix A

# Study of details of Zavlanos algorithms

## A.1 Optimal connection threshold

This appendix will discuss the selection of the optimum threshold ($\xi$) below which a connection will be considered negligible.

The various algorithms do not always return a perfect sparse matrix, and therefore a link is considered inexistent if its absolute value is smaller than some threshold $|a_{ij}| < \xi$. The smaller $\xi$, the more links are included in the estimated network.

In [61] which is a preliminary version of [41], the authors set this threshold to $\xi = 10^{-3}$, but here we will analyze if this value is the optimum one. The equation 3.13 is then modified to the following one:

$$A \in S \Leftrightarrow \begin{cases} a_{ij} \geq \xi & \text{if } s_{ij} = + \\ a_{ij} \leq -\xi & \text{if } s_{ij} = - \\ -\frac{\xi}{2} \geq a_{ij} \geq \frac{\xi}{2} & \text{if } s_{ij} = 0 \\ a_{ij} \in \mathbb{R} & \text{if } s_{ij} = ? \end{cases} \tag{A.1}$$

In table A.1 gives the minimum distance $d$ to the perfect estimation for each algorithm and for different thresholds $\xi$. It can be observed that there is not a huge change in the distance for moderate values of $\xi$. Observing the results we have chose $\xi = 10^{-4}$ for Unstable and SDP algorithm, and $\xi = 10^{-3}$ for the Gersgorin algorithm. It can be observed that $\xi = 10^{-5}$ gives a better result for Gersgorin algorithm, but we think that it is an incidental result because with this small value the algorithm is very sensible to the noise and computational errors and

therefore very unstable.

***Table A.1.*** *Minimum distance d for the different algorithms using the SOS pathway and for different values of the threshold $\xi$*

| $\xi$ | Unstable | Gersgorin | SDP |
|---|---|---|---|
| $10^{-5}$ | 0.67 | 0.57 | 0.67 |
| 1.77e-05 | 0.67 | 0.57 | 0.67 |
| 3.16e-05 | 0.61 | 0.61 | 0.61 |
| 5.62e-05 | 0.62 | 0.62 | 0.62 |
| $10^{-4}$ | 0.56 | 0.60 | 0.56 |
| 0.001 | 0.71 | 0.60 | 0.71 |
| 0.01 | 0.81 | 0.80 | 0.75 |
| 0.1 | 0.80 | 0.86 | 0.91 |

## A.2 L2-Norm

The problem presented in equation 3.12 could be formulated in the following words: "Given steady-state transcription perturbation and mRNA concentration data $U$ and $X$, determine the sparsest stable matrix $A$ that best fits the experimental data which results in a residual $\eta$ small as possible (in some norm), while incorporating any a priori biological knowledge regarding the presence, absence, or nature of specific gene interactions".

Here we present the results of Zavlanos algorithms using a classical Euclidean norm for both datasets, the synthetic data and the data of SOS pathway used in chapter 4.

Therefore, for example the Unstable algorithm presented in equation 3.14, is then transformed to the following convex problem:

$$\begin{aligned} &minimize\ t\sum_{i\neq j} w_{ij}|a_{ij}| + (1-t)\epsilon \\ &subject\ to\ \|AX+BU\|_2 \leq \epsilon, A \in S, \epsilon > 0 \end{aligned} \tag{A.2}$$

The Gersgorin and SDP algorithms are transformed accordingly. All the algorithms still remain convex and therefore are solved with convex optimization methods.

### Synthetic data

The results are obtained in the same way as in subsection 4.2.2. The final results are mean values of the performance metrics over the different synthetic networks.

A summary of the obtained results with the original in-silico dataset is presented in Figure A.1, where the different algorithms of Zavlanos are compared. We can see that with the euclidian norm it is possible to achieve a better result ($d \approx 0.1$) than the one obtained with $l_1$ norm. The best result is obtained with the SDP algorithm and a value of $t = 0.01$, see Figure A.1b.

We have also used the in-silico dataset corrupted with additional Gaussian noise as in subsection 4.2.2, and the results obtained with the euclidean norm are still better than the ones obtained with $l_1$ norm.

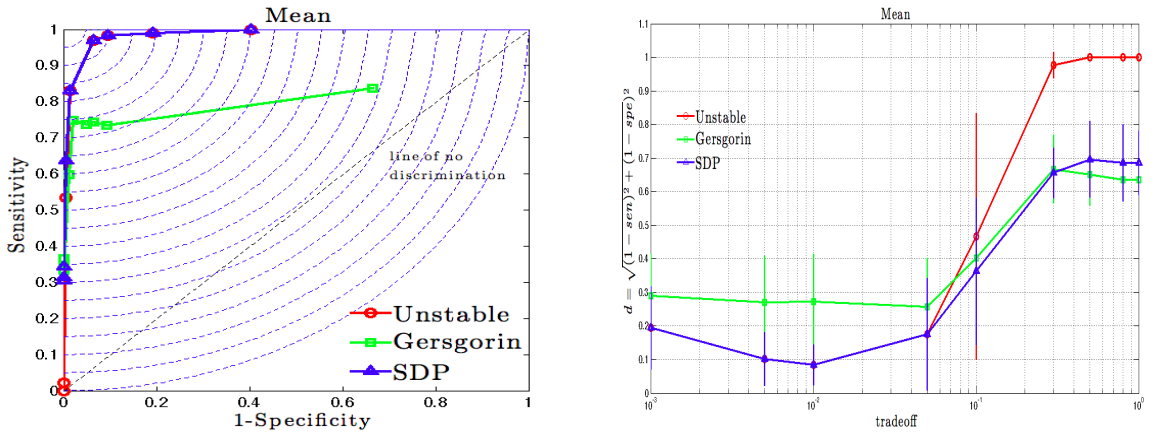



***Figure A.1.*** *(a) ROC plots of modified Zavlanos algorithms for the in-silico dataset, and different values of the parameter t. (b) Distance measure d of the algorithms, in the curves are shown the mean and standard deviation.*

We think that this could happen because the $l_2$ norm is very well suited for Gaussian noise and as the synthetic data is corrupted with this kind of noise, the fit error is better. Therefore, this is a idealistic situation. For example in [62], a model of noise for microarrays which is similar to a mix of normal and log-normal noise is presented. In order see the influence of the noise type, we have used a very similar noise model to corrupt the in-silico data set, and the $l_2$ and $l_1$ norms to measure the error fit. Figure A.2 shows the distance measure $d$ of the Zavlanos algorithms with euclidian norm and $l_1$ norm. It can be observed that both results are very similar and slightly better in the case of using $l_1$ as it is proposed in the original Zavlanos algorithm.

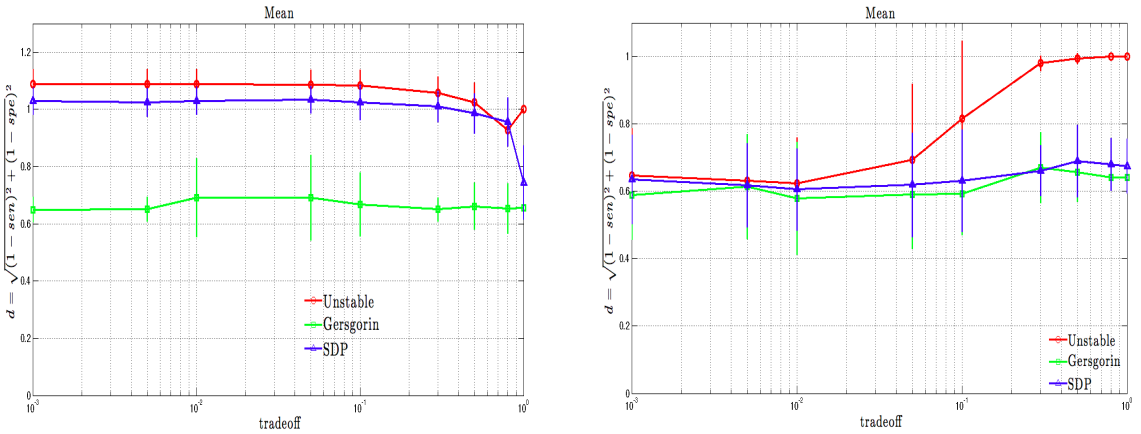



***Figure A.2.*** *Distance comparative plots of Zavlanos algorithms using L2 (a) and L1 (b) norm for in-silico dataset corrupted with additional microarray noise. In the curves the mean and standard deviation are shown.*

### SOS pathway

In order to confirm the results, we have also used the Zavlanos algorithms with $l_2$ norm to measure the errorfit with real data of SOS pathway dataset. The results are presented in Figure A.3, in order to allow an easy comparison the results for l1 norm are reproduced in Figure A.4. In this case, we can see that with the euclidian norm we also obtain worse results that the ones obtained with $l_1$ norm as in the case of synthetic data with microarray noise model. This confirm that the $l_2$ norm is very suited to Gaussian noise but it is not when we have other kind of noise.

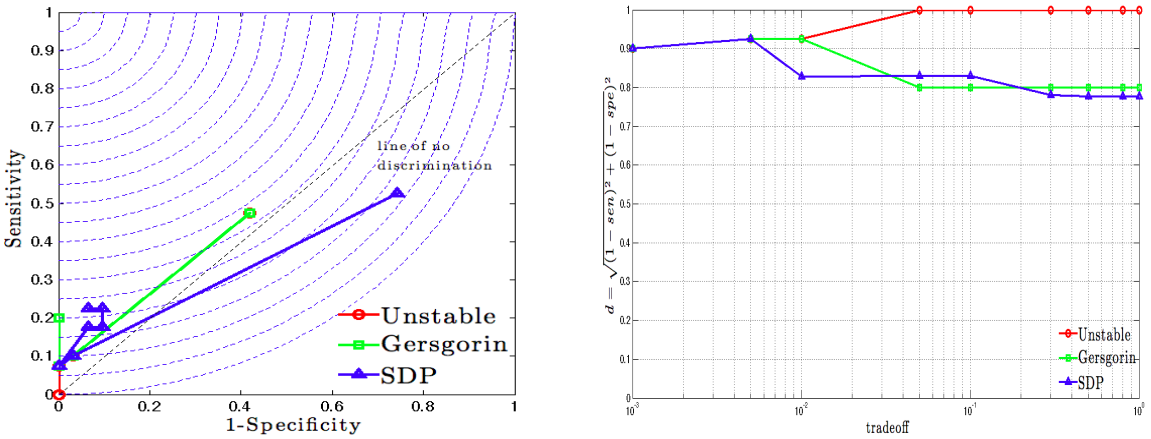



***Figure A.3.*** *(a) ROC plots of modified Zavlanos algorithms for the SOS pathway using L2-norm, and different values of the parameter t. (b) Distance measure d of the algorithms.*

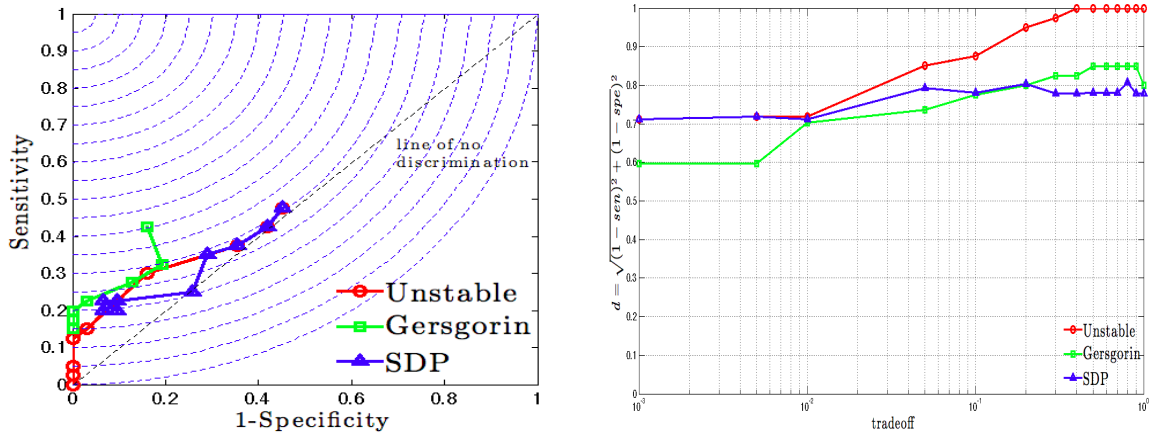



***Figure A.4.*** *(a) ROC plots of Zavlanos algorithms for the SOS pathway using L1-norm, and different values of the parameter t. (b) distance measure d of the algorithms.*

# Bibliography


[1] H. Curtis, S. Barnes, and A. Schnek, *Biologia/ Biology*, Editorial Medica Panamericana Sa de, 7th edition, 2008.

[2] H. Kitano, "Systems biology: a brief overview", *Science*, vol. 295, no. 5560, pp. 1662–1664, 2002.

[3] J. Peto, "Breast cancer susceptibility–a new look at an old model", *Cancer Cell*, vol. 1, no. 5, pp. 411 – 412, 2002.

[4] E. P. Solomon et al., *Biology*, Saunders College Publishing, 4th edition, 1996.

[5] S. Schreiber and E. Lander, "Scanning life's matrix: genes, proteins, and small mollecules", in *The 2002 Holiday Lectures on Science*, 2002.

[6] S. Dudoit and J. Fridlyand, "Bagging to improve the accuracy of a clustering procedure", *Bioinformatics*, vol. 19, no. 9, pp. 1090–1099, 2003.

[7] L. Cardelli, "Abstract machines of systems biology", *Transactions on Computational Systems Biology III*, pp. 145–168, 2005.

[8] J. Monod, "From enzymatic adaptation to allosteric transitions", *Science*, vol. 154, no. 3748, pp. 475, 1966.

[9] D. Marbach, *Evolutionary reverse engineering of gene networks*, PhD thesis, École Polytechnique Fédérale de Lausanne, Lausanne, 2009.

[10] H. Bolouri and E.H. Davidson, "Modeling transcriptional regulatory networks", *BioEssays*, vol. 24, no. 12, pp. 1118–1129, 2002.

[11] J.M. Bower and H. Bolouri, *Computational modeling of genetic and biochemical networks*, The MIT Press, 2004.

[12] D.F. Browning and S.J.W. Busby, "The regulation of bacterial transcription initiation", *Nature Reviews Microbiology*, vol. 2, no. 1, pp. 57–65, 2004.

[13] M. Thattai, *The dynamics of genetic networks*, PhD thesis, Massachusetts Institute of Technology, Department of Physics, 2004.

[14] D. di Bernardo, T. S. Gardner, and J. J. Collins, "Robust identification of large genetic networks", *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pp. 486–497, 2004, PMID: 14992527.

[15] M. Bansal, V. Belcastro, A. Ambesi-Impiombato, and D. di Bernardo, "How to infer gene networks from expression profiles", *Molecular Systems Biology*, vol. 3, no. 1, Feb. 2007.

[16] A. Ambesi-Impiombato and D. di Bernardo, "Computational biology and drug discovery: From single-target to network drugs", *Current Bioinformatics*, vol. 1, no. 1, pp. 3–13, 2006.

[17] M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns", *Proceedings of the National Academy of Sciences*, vol. 95, no. 25, pp. 14863, 1998.

[18] M. Bosio, P. Bellot Pujalte, P. Salembier, and A. Oliveras-Verges, "Feature set enhancement via hierarchical clustering for microarray classification", in *Genomic Signal Processing and Statistics (GENSIPS), 2011 IEEE International Workshop on*. IEEE, 2011, pp. 226–229.

[19] P. Bellot Pujalte, "Study of gene expression representation with treelets and hierarchical clustering algorithms", *Universitat Politècnica de Catalunya*, 2011, Proyecto final de carrera.

[20] A.B. Lee, B. Nadler, and L. Wasserman, "Treelets–an adaptive multi-scale basis for sparse unordered data", *The Annals of Applied Statistics*, vol. 2, no. 2, pp. 435–471, 2008.

[21] R. Amato, A. Ciaramella, N. Deniskina, C. Del Mondo, D. di Bernardo, C. Donalek, G. Longo, G. Mangano, G. Miele, G. Raiconi, et al., "A multi-step approach to time series analysis and gene expression clustering", *Bioinformatics*, vol. 22, no. 5, pp. 589–596, 2006.

[22] R.J. Prill, D. Marbach, J. Saez-Rodriguez, P.K. Sorger, L.G. Alexopoulos, X. Xue, N.D. Clarke, G. Altan-Bonnet, and G. Stolovitzky, "Towards a rigorous assessment of systems biology models: the dream3 challenges", *PloS one*, vol. 5, no. 2, pp. e9202, 2010.

[23] M.I. Arnone and E.H. Davidson, "The hardwiring of development: organization and function of genomic regulatory systems", *Development*, vol. 124, no. 10, pp. 1851–1864, 1997.

[24] D. Thieffry, A.M. Huerta, E. Pérez-Rueda, and J. Collado-Vides, "From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in escherichia coli", *Bioessays*, vol. 20, no. 5, pp. 433–440, 1998.

[25] A.J. Butte and I.S. Kohane, "Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements", in *Pac Symp Biocomput*, 2000, vol. 5, pp. 418–429.

[26] R. Steuer, J. Kurths, C.O. Daub, J. Weise, and J. Selbig, "The mutual information: detecting and evaluating dependencies between variables", *Bioinformatics*, vol. 18, no. suppl 2, pp. S231–S240, 2002.

[27] D.M. Chickering, "Learning bayesian networks is np-complete", *Lecture Notes in Statistics-new york-springer verlag-*, pp. 121–130, 1996.

[28] S.J. Russell and P. Norvig, *Artificial intelligence: a modern approach*, Prentice hall, 2010.

[29] D. Heckerman et al., "A tutorial on learning with bayesian networks", *Nato Asi Series D Behavioural And Social Sciences*, vol. 89, pp. 301–354, 1998.

[30] N. Friedman, M. Goldszmidt, and A. Wyner, "Data analysis with bayesian networks: A bootstrap approach", in *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 1999, pp. 196–205.

[31] D.H. Wolpert and W.G. Macready, "No free lunch theorems for optimization", *Evolutionary Computation, IEEE Transactions on*, vol. 1, no. 1, pp. 67–82, 1997.

[32] J. McEntyre and D. Lipman, "Pubmed: bridging the information gap", *Canadian Medical Association Journal*, vol. 164, no. 9, pp. 1317–1319, 2001.

[33] K.M. Frizzell, M.J. Gamble, J.G. Berrocal, T. Zhang, R. Krishnakumar, Y. Cen, A.A. Sauve, and W.L. Kraus, "Global analysis of transcriptional regulation by poly (adp-ribose) polymerase-1 and poly (adp-ribose) glycohydrolase in mcf-7 human breast cancer cells", *Journal of Biological Chemistry*, vol. 284, no. 49, pp. 33926, 2009.

[34] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using bayesian networks to analyze expression data", *Journal of computational biology*, vol. 7, no. 3-4, pp. 601–620, 2000.

[35] A. Djebbari and J. Quackenbush, "Seeded bayesian networks: constructing genetic networks from microarray data", *BMC systems biology*, vol. 2, no. 1, pp. 57, 2008.

[36] H. De Jong, "Modeling and simulation of genetic regulatory systems: a literature review", *Journal of computational biology*, vol. 9, no. 1, pp. 67–103, 2002.

[37] M. Bansal, D. di Bernardo, et al., "Inference of gene networks from temporal gene expression profiles", *IET systems biology*, vol. 1, no. 5, pp. 306–312, 2007.

[38] J. Hertz, A. Krogh, and R.G. Palmer, *Introduction to the theory of neural computation*, vol. 1, Westview press, 1991.

[39] T.S. Gardner, D. di Bernardo, D. Lorenz, and J.J. Collins, "Inferring genetic networks and identifying compound mode of action via expression profiling", *Science's STKE*, vol. 301, no. 5629, pp. 102, 2003.

[40] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*, vol. 1, Springer Series in Statistics, 2001.

[41] M.M. Zavlanos, A.A. Julius, S.P. Boyd, and G.J. Pappas, "Inferring stable genetic networks from steady-state data", *Automatica*, 2011.

[42] S.P. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge Univ Pr, 2004.

[43] S.P Boyd, "$l_1$–norm methods for convex cardinality problems. Lecture notes for EE364b. Stanford University", http://www.stanford.edu/class/ee364b/, 2011, [Online; accessed 18-July-2012].

[44] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21", http://cvxr.com/cvx/, Apr. 2011.

[45] C.H. Edwards and D.E. Penney, *Elementary Differential Equations with Boundary Value Problems*, Prentice Hall, 6 edition, Dec. 2007.

[46] R.A. Horn and C.R. Johnson, *Matrix analysis*, Cambridge Univ Pr, 1990.

[47] Z. Gajić and M.T.J. Qureshi, *Lyapunov matrix equation in system stability and control*, vol. 195, Academic Pr, 1995.

[48] G. Stolovitzky, D. Monroe, and A. Califano, "Dialogue on reverse-engineering assessment and methods", *Annals of the New York Academy of Sciences*, vol. 1115, no. 1, pp. 1–22, 2007.

[49] Wikipedia, "Receiver operating characteristic — Wikipedia, the free encyclopedia", http://en.wikipedia.org/wiki/Receiver_operating_characteristic, 2012, [Online; accessed 19-July-2012].

[50] H. Hache, H. Lehrach, and R. Herwig, "Reverse engineering of gene regulatory networks: a comparative study", *EURASIP Journal on Bioinformatics and Systems Biology*, vol. 2009, pp. 8, 2009.

[51] J. Surowiecki, *The Wisdom of Crowds*, Anchor Books. Knopf Doubleday Publishing Group, 2005.

[52] D. Marbach, C. Mattiussi, and D. Floreano, "Combining multiple results of a reverse-engineering algorithm: Application to the DREAM five-gene network challenge", *Annals of the New York Academy of Sciences*, vol. 1158, no. 1, pp. 102–113, Mar. 2009.

[53] T. Dietterich, "Ensemble methods in machine learning", in *Multiple Classifier Systems*, vol. 1857 of *Lecture Notes in Computer Science*, pp. 1–15. Springer Berlin / Heidelberg, 2000.

[54] S.L. Schreiber, "Target-oriented and diversity-oriented organic synthesis in drug discovery", *Science*, vol. 287, no. 5460, pp. 1964–1969, 2000.

[55] D. Marbach, R. J. Prill, T. Schaffter, C. Mattiussi, D. Floreano, and G. Stolovitzky, "Revealing strengths and weaknesses of methods for gene network inference", *Proceedings of the National Academy of Sciences*, vol. 107, no. 14, pp. 6286–6291, Mar. 2010.

[56] A. Mochizuki and D. Saito, "Analyzing steady states of dynamics of bio-molecules from the structure of regulatory networks", *Journal of Theoretical Biology*, vol. 266, no. 2, pp. 323–335, Sept. 2010.

[57] M. Bansal, G. Della Gatta, and D. di Bernardo, "Inference of gene regulatory networks and compound mode of action from time course gene expression profiles", *Bioinformatics*, vol. 22, no. 7, pp. 815–822, 2006.

[58] T. J. Perkins, "The gap gene system of drosophila melanogaster", *Annals of the New York Academy of Sciences*, vol. 1115, no. 1, pp. 116–131, 2007.

[59] Kevin Y. Yip, Roger P. Alexander, Koon-Kiu Yan, and Mark Gerstein, "Improved reconstruction of in silico gene regulatory networks by integrating knockout and perturbation data", *PLoS ONE*, vol. 5, no. 1, pp. e8121, Jan. 2010.

[60] J. Tegnér, M. K. Stephen Yeung, J. Hasty, and J. J Collins, "Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling", *Proceedings of the National Academy of Sciences*, vol. 100, no. 10, pp. 5944–5949, May 2003.

[61] A. Julius, M. Zavlanos, S. Boyd, and G.J. Pappas, "Genetic network identification using convex programming", *Systems Biology, IET*, vol. 3, no. 3, pp. 155 –166, May 2009.

[62] Y. Tu, G. Stolovitzky, and U. Klein, "Quantitative noise analysis for gene expression microarray experiments", *Proceedings of the National Academy of Sciences*, vol. 99, no. 22, pp. 14031–14036, Oct. 2002.