Biological network inference via DTW & Correlation measures from time-course data
## Contents

1 Mathematical background 11
   1.1 Similarity measures for longitudinal data 11
       1.1.1 Longitudinal data (or time series) 11
       1.1.2 Correlation 13
       1.1.3 DTW: Dynamic Time Warping 13
   1.2 Network Theory 21
       1.2.1 Principles 21
       1.2.2 Random and Regular networks 23
       1.2.3 Small-world and Scale-free networks 24
       1.2.4 Network distances 26

2 Network inference background 33
   2.1 Networks in Cell Biology 33
   2.2 State-of-the-art 35
   2.3 Correlation & DTW-based new method 36
       2.3.1 Building the optimal correlation and DTW matrices 40
       2.3.2 New combining method 41
       2.3.3 Graphs visualization 41

3 Results on general purpose synthetic data 43
   3.1 Synthetic data 43
       3.1.1 Adjacency matrix and simulated biological time series 43
       3.1.2 Simulation of co-regulated genes 44
   3.2 Experimental design and results 45
   3.3 Discussion 57

4 Results on biologically-like synthetic data 59
   4.1 GeneNetWeaver (GNW) 59
       4.1.1 The GNW tool for simulation of regulatory networks 59
       4.1.2 Synthetic data generation 60
   4.2 Improved algorithms for inferring regulatory networks 60
       4.2.1 Issues with perturbation-based simulated data 61
       4.2.2 First method 61
CONTENTS

4.2.3 Second method .............................. 62
4.3 Results ........................................... 62
   4.3.1 First method results ..................... 62
   4.3.2 Second method results .................. 64
4.4 Discussion ................................. 64

5 Results on biological data ......................... 67
   5.1 SOS DNA repair system of Escherichia coli ........ 67
      5.1.1 Results ................................... 68
   5.2 Arabidopsis thaliana system .................. 68
      5.2.1 Results ................................... 71
   5.3 Discussion ................................. 74

6 Summary and Conclusions ......................... 75

Bibliography ................................ 77
Acknowledgments

It is a great pleasure to thank my supervisor, Dr. Giuseppe Jurman, for having offered me the opportunity of coming to the Fondazione Bruno Kessler and working on an exciting research area. I equally thank Samantha Riccadonna for having followed me during these five months and helped me out whenever difficulties arose. You both have shown me how challenging and rewarding the profession of researcher could be.

I thank Prof. Cesare Furlanello for welcoming me in his group, for the availability he has always shown and for guiding me while taking important decisions for my next career steps. I also want to thank the whole MPBA (Predictive Models for Biomedicine & Environment) group for allowing me to have a very pleasant time each day at work and for all the support. Thank you.

A particular thanks is for Prof. Pugliese Andrea and Prof. Iannelli Mimmo, for always being present and helpful in times of need. Their commitment and passion have pushed me further in discovering the beautiful world of mathematics applied to biology. I hope this area will continue to grow among young mathematicians. Thank you.

I am especially grateful to Prof. Luisa Malaguti (DiSMI, University of Modena and Reggio Emilia) for having followed me since my bachelor studies. You were able to create a special relationship of trust, as only a good professor can do. Thank you.

Passing now to my mother tongue,

Ringrazio le Professoressa Paola Bergamini e Elisabetta Fiorini (Liceo Scientifico M. Fanti, Carpi), Insegnanti con la I maiuscola, per avermi trasmesso la passione per la matematica, la biologia e le scienze in genere, basi solide sul quali ho potuto lavorare e creare tutto ciò che ha portato a questo elaborato finale. Grazie di cuore e continue ad insegnare con la stessa passione.
Grazie, Davide Zoboli, per essermi amico ormai da quasi tre anni. Per le litigate, le incomprensioni e le riappacificazioni. Per le serate passate a chiacchierare davanti ad un biliardo o dal piadinare. Per i pomeriggi passati ad allenarci insieme o in piscina. Per i giri in macchina, le storie raccontate e i consigli ricevuti. Per avermi raccontato così tanto di te e allo stesso tempo sapere così tanto di me. Grazie Davide.

Un grazie speciale per la persona che sopporta i miei difetti quotidianamente, con grande forza di volontà. Grazie per scegliere, giorno dopo giorno, di vivere insieme a me. Grazie per consigliarmi ed ascoltarmi in ogni momento. Grazie per gli splendidi momenti passati, presenti e futuri della nostra vita. Ti sei fidata di me quando avresti potuto scegliere strade più semplici. So di potermi fidare di te. Sai farmi ridere quando nessun altro ci riuscirebbe, grazie di cuore Damiana.

Grazie nonna Assunta, per essere sempre stata una persona presente nella famiglia. Sei un’esempio di personalità forte e voglia di vivere. Rimarranno per sempre nel mio cuore le tue parole al telefono: un’abbracciatona forte forte nanì...! Grazie di cuore nonna.

Non ci sono parole sufficientemente appropriate per ringraziare mio fratello Luca Ferrarini, per il continuo ed instancabile sostegno donatomi. Senza il tuo prezioso aiuto non sarei mai riuscito a rendere così speciali i percorsi di vita che ho intrapreso negli anni. So che continuerai ad essere sempre presente, e di questo ti sarò eternamente grato. Mi sei stato vicino in ogni occasione, come fratello, come compagno e come indubbio insegnante. Grazie di cuore Luca.

Dopo aver pensato per settimane al metodo migliore in cui scrivere le prossime righe di ringraziamenti, tutto ciò che effettivamente penso possa valere è il mio più sincero GRAZIE mamma e papà (Barletta Valeria e Ferrarini Claudio). Non potrei elencare tutto ciò per cui vorrei e dovrei ringraziarvi. Siete unici. Grazie di cuore, per tutto.
Introduction

The main objective of this thesis is to investigate new methods to infer gene regulatory networks. Genes are the units of inheritance of living organisms and play a fundamental role in building and maintaining cells. They are concatenated in a double helix molecule called DNA, whose fragments are packed into macro-structures referred to as chromosomes, which in turn are all contained in each organism cell’s nucleus. For example, human DNA is packed in 46 chromosomes, and contains approximately 25000 protein-coding genes [1]. Throughout the cell live, genes are first transcribed into messenger RNA (mRNA) which exits the nucleus and it is further translated into proteins. These are biological structures consisting of polypeptides with a typically globular or fibrous shape which are responsible for the correct functioning of the cell. Some of these proteins have the capability of entering the nucleus and activate (or inhibit) the transcription process for a given gene: such proteins are called transcription factors (TF) [2].

Different biological techniques have been developed in order to analyze changes in gene expression levels. One of the most widely used techniques is the one based on microarrays [3]. These chips present a collections of DNA probes (short sections of a gene used to hybridize complementary RNA or cDNA samples called targets) installed on a thin layer of glass, plastic or silicon biochip. Messenger RNA is first extracted from the cell, bound to fluorophores (fluorescent chemical compound that re-emit light upon light excitation), and inserted in the array; the mRNA hybridizes to the different probes, with the quantity of hybridization being proportional to the quantity of the specific mRNA present in the biological sample. Lasers are then used to acquire images of the microarray: each probe will provide a different signal, with intensity proportional to the amount of hybridization. Eventually, one can derive the level of expression of a given gene from the intensity of the corresponding probe. In order to investigate how a biological system responds to stimuli, one can set up a series of experiments in which mRNA is extracted from different biological conditions (and in different moments in time): microarray analysis will therefore result in a set of expression levels for each gene: these values can be thought of as a temporal series, showing how the expression of a given gene changes due to changes in biological conditions (and in time). More information can be found in [4], [5], [6] and
The interactions between regulatory genes and regulated ones form what is usually referred to as gene regulatory network. Reverse engineering of gene regulatory networks (or network inference) has become increasingly more important in the recent years within the framework of Systems biology. Nowadays, there is a great interest in modeling biological systems, and in particular the system of transcriptional regulations. Complex network theory represent an excellent mathematical framework to investigate such research field. Mathematical structures called graphs are used in order to model the interactions between genes, with the latter being modeled by nodes, whereas interactions between genes are modeled by links between nodes. The real interplays between TF’s and regulated genes should be modeled by weighted directed links, since the flow of information between them it is not necessarily bi-directional, and the strength of interaction changes depending on the involved genes. However, within this work, we will focus on a simplified problem: namely, that of modeling gene regulatory networks with undirected and unweighted graphs.

In fact, the goal of our work is to infer such undirected networks starting from the time series obtained by microarrays. The hypothesis is that genes which are biologically ‘related’ will present mathematically related signals. In order to discover such relations, several methods have been introduced in literature. We will focus on those based on Correlation-like measures. Examples of these methods include linear correlations (e.g., Pearson, Spearman,...), dynamic time warping (DTW) and others [8]. Taken separately, each of these methods presents some limitations. For example, Pearson’s correlation lacks sensitivity in case of non-linear relations between signals (e.g. when signals are delayed with respect to each other); conversely, the DTW presents limitations when signals, although related, present different levels of expression. What we will do is to investigate the possibility to combine different correlation-like measures (i.e., Pearson correlation and DTW) in order to overcome these limitations.

In order to develop new inferring methods, we first need to simulate biological data which will later be used as ground truth for validation purposes. Such initial validation set will be based on simple biological assumptions, such as delay between regulated and regulating gene and noise. After having developed and tested our new inferring method on this data, we will apply it, eventually improving it as needed, to more realistic simulation of microarray data: these will be obtained by an open-source tool (GeneNetWeaver, GNW) for in-silico benchmark generation. To conclude, in order to complete our analysis, we will test our methods on real biological data.

The structure of the thesis is as follows: in chapter 1 we will present the mathematical background needed in which our work is rooted. Fundamental concepts such as Correlation, DTW and Networks theory will be introduced. In chapters 3, 4 and 5, we will introduce and further improve our inferring
methods, applying them to the synthetic and real data mentioned before: each chapter will include the description of the experiments and the corresponding results. Finally, we will summarize the overall analyses in the conclusions.
Chapter 1

Mathematical background

In this chapter we provide the mathematical background which will serve as the theoretical basis for the developed methods. In the first part, we will introduce the Similarity measures, among which we can find Correlation and Dynamic Time Warping (DTW). Complex networks analysis and network inference are the mathematical tools introduced in the second part of this chapter. Some distances for network comparison such as Hamming distance, Ipsen-Mikhailov distance and HIM distance will also be studied.

1.1 Similarity measures for longitudinal data

In recent years various mathematical approaches were created in order to determine how much two signals are related. Among them, we cite the family of the correlation measures. Many variants of these measures are described in literature, such as [9]. We will focus on Pearson’s correlation and on Dynamic Time Warping.

1.1.1 Longitudinal data (or time series)

Formally speaking, a longitudinal data (or time series) is a sequence of data points, measured at successive time instants. It represents the same type of information on the same subject at different points in time (see Figure 1.1). For example, considering a scientific journal, the number of its annual citations can be seen as a time series. The theory which study these kind of data is the Time Series Analysis; it focuses on methods for analyzing longitudinal data, extracting meaningful statistics from them. For deeper information, the reader is referred to [10].

1.1.2 Correlation

The correlation is one of the most useful statistics in order to detect ‘similarities’ between signals; it describes the degree of dependency between two
variables. Dependence refers to any situation in which random variables do not satisfy the probabilistic independence. Formally, given two random variables $X$ and $Y$, we can define their cumulative distribution functions as

$$F_X(x) = P(X \leq x) \quad \text{and} \quad F_Y(y) = P(Y \leq y).$$

In particular, for continuous random variables, we have

$$F_X(x) = \int_{-\infty}^{x} f(t)dt \quad \text{and} \quad F_Y(y) = \int_{-\infty}^{y} g(s)ds$$

where $f$ and $g$ are the probability density functions. We can now define the joint cumulative distribution of the variables $X$ and $Y$ as

$$F(x, y) = P(X \leq x, Y \leq y).$$

Thus, we have that $X$ and $Y$ are independent if and only if $F(x, y) = F_X(x)F_Y(y)$.

Among the correlation coefficients, the most common one is the Pearson’s coefficient, which is mostly sensitive to linear relationships between variables. It is obtained by dividing the covariance of the two variables by the product of their standard deviations as in 1.1:

$$\rho_{X,Y} = \text{corr}(X,Y) = \frac{\text{Cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{\mathbb{E}[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}. \quad (1.1)$$
The quantity in 1.1 is defined only if both standard deviations are finite and nonzero; furthermore, the correlation turns out to be in the range of values \([-1, 1]\). The coefficient has value \(\rho_{X,Y} = 1\) in case of a perfect positive linear relationship, and \(\rho_{X,Y} = -1\) in case of a perfect negative linear relationship (anticorrelation). As the two signals are increasingly less related, the coefficient gets closer to zero. Examples of Pearson’s correlation coefficients are illustrated in Figures 1.1.

**MIC: Maximal Information Coefficient**

As previously mentioned, correlation lacks sensitivity in detecting non-linear relations; in order to improve the capabilities of the correlation, it has recently been created a new measure: it is called **MIC** (Maximal Information Coefficient) and it provides a quick way to evaluate (non-linear) associations between lots of variables [11]. In particular, thinking of building gene regulatory predictive models, we could use it to evaluate potential predictors. When it comes to capturing non-linear relationships, the MIC is superior to the correlation coefficient and Mutual Information. On the other hand, it provides similar results when faced with linear associations.

Returning to the correlation weaknesses, there is a lack which we have have considered more than others: correlation is limited in detecting possible shifts along the time axis. Such limitation led us to introduce another tool in order to investigate similarities between genetic profiles: Dynamic Time Warping.

**1.1.3 DTW: Dynamic Time Warping**

Dynamic Time Warping (DTW, for short) is a measure of similarity between two sequences which may vary in time. In general, it allows to find an optimal match between the two given series: they are ‘warped’ non-linearly in the time dimension to determine a measure of their similarity, stretching (or compressing) the time axis. As a comprehensive reference, the reader is referred to [12].

DTW has been primarily used to compare different speech patterns in automatic speech recognition [13], but it has been also used in other important research area [14]. In our work, we apply the DTW to biological time series. Particularly, we use the DTW in order to re-align genetic profiles prior to evaluate their correlation. This can be useful when biological processes are linearly correlated but display time delays. Hereafter we show a working definition of the DTW algorithm.

Let us consider two time series A (of length n) and B (of length m), where

\[
A = a_1, ..., a_t, ..., a_n
\]
\[
B = b_1, ..., b_j, ..., b_m.
\]
### CHAPTER 1. MATHEMATICAL BACKGROUND

<table>
<thead>
<tr>
<th>Relationship Type</th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$\rho(f_1, f_2)$</th>
<th>plot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uncorrelated</td>
<td>$\sin(x)$</td>
<td>$\cos(x)$</td>
<td>$-0.0015$</td>
<td><img src="image1.png" alt="plot" /></td>
</tr>
<tr>
<td>Positively correlated</td>
<td>$\sin(x)$</td>
<td>$\sin(x) - 0.3$</td>
<td>$1$</td>
<td><img src="image2.png" alt="plot" /></td>
</tr>
<tr>
<td>Negatively correlated</td>
<td>$\sin(x)$</td>
<td>$-\sin(x)$</td>
<td>$-1$</td>
<td><img src="image3.png" alt="plot" /></td>
</tr>
</tbody>
</table>

Table 1.1: Examples of Pearson’s coefficient. Black: $f_1$. Red: $f_2$. All functions are plotted in the interval $[0, 2\pi]$ for 32 points.
In order to align them using DTW, one first needs to construct an \( n \times m \) matrix \( M \) whose generic element \( m_{ij} \) represents a local distance between the two time points \( a_i \) and \( b_j \) (generally the Euclidean distance \( d(a_i, b_j) = (a_i - b_j)^2 \) is used). Thus, each \( m_{ij} \) corresponds to the alignment between \( a_i \) and \( b_j \). We can now define a warping path \( W \) as a set of matrix (or grid) elements which defines a map between \( A \) and \( B \). The \( k^{th} \) element of \( W \) is \( w_k = (i, j)_k \): \( W = w_1, ..., w_k, ..., w_K \) where \( \max(m, n) \leq K < (m + n - 1) \). These two extremes are the minimum and the maximum possible paths within an \( m \)-by-\( n \) grid in which you have to start from the position \((1,1)\) and you have to arrive to the position \((m,n)\). Generally the path is subject to several constraints [15], such as:

- boundary conditions: \( w_1 = (1,1) \) and \( w_K = (m, n) \).

- continuity: given \( w_k = (x, y) \), then \( w_{k-1} = (a, b) \) where \( x - a \leq 1 \) and \( y - b \leq 1 \). In this way, we restrict the allowable steps to adjacent cells.

- monotonicity: given \( w_k = (x, y) \), then \( w_{k-1} = (a, b) \) where \( x - a \geq 0 \) and \( y - b \geq 0 \).

For a better understanding, one needs to introduce the concept of Dynamic Programming Matrix (DPM) [12]. The DPMs are plots (on a grid) of the alignment paths obtained for the two time series: on the x-axis we have the time points of the first signal, while on the y-axis there are the time points of the second one (for more information, the reader is referred to [16, 17, 15]).

The distance (which can exceed the interval \([0,1]\)) between the two signals is computed by adding the distances of individual aligned elements [18]. Figures 1.3-1.7 show the application of the DTW to different curves:

- Figure 1.3: plot of two perfect aligned curves (\( \sin(x) \) and \( \sin(x) \))

- Figure 1.4: plot of \( \sin(x) \) and \(-\sin(x)\); the first half of \(-\sin(x)\) is recognized as a time shift of the second half of \( \sin(x) \)

- Figure 1.5: plot of \( \sin(x) \) and \( \sin(x + \frac{\pi}{2}) \); the first three quarters of \( \sin(x + \frac{\pi}{2}) \) are recognized as a time shift of the second, third and fourth quarters of \( \sin(x) \)
• Figure 1.6: plot of $\cos(x)$ and $-\cos(x)$; as for Figure 1.4, the first half of $-\cos(x)$ is recognized as a time shift of the second half of $\cos(x)$

• Figure 1.7: plot of $\cos(x)$ and $\sin(x)\cos(x)$; it becomes more difficult for the DTW to find a unique possible time shift, as it can be seen in the plot of the time series alignment (Figure 1.7 (b))

For this work we made use of $R$, a language and environment for statistical computing and graphics, and in particular of theirs packages (sets of R functions and compiled code in a well-defined format) [19],[20]. Using the dtw package, it is possible to detect the best shift between two time series, thanks to two members of the dtw class: index1 and index2. The first one is a vector containing all (and only) the time series indices of the points (natural numbers) from which the lines of the DPM start (these lines are shown as gray dotted lines in figures 1.3-1.7). The other vector, instead, contains all (and only) the time series indices of the points (natural numbers) to which the lines arrive.

In order to better understand, let us consider the example in Table 1.2. Subtracting Index1 from Index2, we obtain the vector $v$ (shown in the same caption) whose elements are the shifts suggested by dtw process (point by point).

![Figure 1.2](image)

Figure 1.2: (a) Red line: $\sin(x)$. Black line: $\cos(x)$.
$\text{Cor}(\sin(x), \cos(x)) = -0.0014963$.
(b) Green line: shifted $\sin(x)$. Black line: $\cos(x)$.
$\text{Cor}(\text{shifted}\_\sin(x), \cos(x)) = 0.1563970$. This new correlation value is the one that should be used in the creation of the inferred network.

The mode of $v$ ($\text{mode}(v) = -8$) represents the estimate of the shift that better could make similar the second series to the first one. In this case, the $\sin(x)$ should be shift to the left of 8 steps (see Figure 1.2). This is the idea
1.1. SIMILARITY MEASURES FOR LONGITUDINAL DATA

Table 1.2: (a) Red line: sin(x). Black line: cos(x). (b) Table with Index1, Index2 and v.
underlying the method presented in Chapter 3.

Figure 1.3: (a) Plot of the DTW alignment of $\sin(x)$ over itself. (b) Dynamic programming matrix of the alignment. The DTW distance among these two series is 0.

Figure 1.4: (a) Plot of the DTW alignment of $\sin(x)$ over $-\sin(x)$. (b) Dynamic programming matrix of the alignment. The DTW distance among these two series is 82.75325
1.1. SIMILARITY MEASURES FOR LONGITUDINAL DATA

Figure 1.5: (a) Plot of the DTW alignment of $\sin(x)$ over $\sin(x + \left(\frac{\pi}{2}\right))$. (b) Dynamic programming matrix of the alignment. The DTW distance among these two series is 35.28178.

Figure 1.6: (a) Plot of the DTW alignment of $\cos(x)$ over $-\cos(x)$. (b) Dynamic programming matrix of the alignment. The DTW distance among these two series is 126.9751.
Figure 1.7: (a) Plot of the DTW alignment of $\cos(x)$ over $\sin(x)\cos(x)$. (b) Dynamic programming matrix of the alignment. The DTW distance among these two series is 72.13446.
1.2 Network Theory

This section provides the basic concepts of the network theory. We first introduce graphs and some of their properties, such as clustering coefficient and characteristic path. Then we pass to the introduction of some special network topologies and finally to the notion of network distance [21].

1.2.1 Principles

A network (or graph) is defined as a set of nodes (or vertices) connected by edges (or links). Links can be either directed or undirected. A directed link starts from a source node and ends in a target node, the direction indicating the type of relationship which exists between the two nodes. Conversely, an undirected link joins two nodes without specifying a source nor a target [21].

Formally, any network can be represented as a graph, a mathematical entity consisting of \( N \) nodes (vertices) and \( E \) edges (links or arrows) connecting pairs of nodes and representing interactions (\( N \in \mathbb{N} \cup \{\infty\} \)). Loops are allowed, i.e. an edge can link the same node to indicate self-interaction (some authors use the term pseudograph to indicate graph with loops). Edges can be bidirectional or unidirectional: in the latter case the graph is called directed (digraph, for short) and the edges are represented by arrows [22]. Moreover, edges can carry weights to indicate interaction intensity: in this case, the network is called weighted. More refined structures exist but they are not considered here. For instance: labeled graphs, where functions from some subsets of the integers to the vertices (edges) of the graph identify classes of vertices (edges); hypergraphs, where an edge can connect any number of vertices; and multigraphs, where any numbers of edges between two vertices are allowed. For any network \( G \), its topology consists of the set \( V(G) = \{v_1, \ldots, v_N\} \) of its nodes and the set \( E(G) = \{e_1 = (v_{i_1}, v_{j_1}), \ldots, e_E = (v_{i_E}, v_{j_E})\} \) of its edges, neglecting weights and directions. Different types of graph sharing the same topology are displayed in Fig. 1.8.

A network, or graph, is characterized completely by its adjacency matrix \( A \), i.e. an \( N \times N \) matrix whose nonzero entries denote the various links.
between the graph’s \( N \) nodes. Directions and weights are represented by the signs (or by asymmetricity) and values of the matrix entries. For the underlying topology (and thus for any unweighted undirected network), the adjacency matrix is symmetric and with entries in \( \{0,1\} \). The adjacency matrices for the weighted digraph in Fig. 1.8 and its topology are shown in Tab. 1.3, where nodes ordering is clockwise starting from the top node. This representation is not unique, in that it depends on the actual labeling of the nodes, and isomorphic graphs (identical graphs with permuted labels) share the same adjacency matrix. Similarly, graphical representations are not unique too, since node placement is arbitrary.

The degree (\( \text{deg} \)) of a vertex in an undirected graph is the number of edges touching the vertex itself, with loops (usually, but not for all authors) counted twice. The degree matrix is the diagonal matrix with the vertex degrees: for instance, for the network topology in Fig. 1.10, the degree matrix is \( D = \begin{pmatrix} 2 & 4 & 1 & 3 \\ 1 & 1 & 1 & 3 \end{pmatrix} \). The Laplacian matrix of a graph is defined as the difference between the degree and the adjacency matrices: \( L = D - A \). Thus, for an undirected and unweighted graph with no loops (a simple graphs), \( L \) has zero row/column sum.

There exist at least two different normalized versions of the Laplacian matrix, namely \( \mathcal{L} = D^{-\frac{1}{2}} LD^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \) and \( \Delta = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} \), where \( I \) is the identity matrix and \( D^{-\frac{1}{2}} \) is the diagonal matrix with entries \( \frac{1}{\sqrt{\text{deg}_i}} \).

In terms of the degree, their entries can be explicitly written as:

\[
\mathcal{L} = \begin{cases} 1 & \text{if } i = j \text{ and } \text{deg}_i \neq 0 \\ -\frac{1}{\sqrt{\text{deg}_i \text{deg}_j}} & \text{if } ij \text{ is an edge} \\ 0 & \text{otherwise} \end{cases}
\]

\[
\Delta = \begin{cases} 1 & \text{if } i = j \text{ and } \text{deg}_i \neq 0 \\ -\frac{1}{\text{deg}_j} & \text{if } ij \text{ is an edge} \\ 0 & \text{otherwise} \end{cases}
\]

The matrices \( \mathcal{L} \) and \( \Delta \) are similar so they have the same set of eigenvalues (spectrum).

The matrices \( A, L, \mathcal{L} \) and \( \Delta \) are called connectivity matrices of the graph [23].

The study of these structures is one of the fundamental pillars of discrete mathematics. Important applications of graph theory are found in several scientific fields, such as social sciences, business relations, computer science and biology [24]. Examples for the last field include networks modeling epidemiological phenomena, neuronal functional and anatomical connections, metabolic pathways, protein-protein interactions and gene regulatory interactions [25, 26]. The application of graph theory to the study of regulatory network is at core of this research.

During recent years, research interest has shifted from studying the properties of single nodes and links to investigating larger-scale statistical prop-
1.2. NETWORK THEORY

properties of whole complex graphs, such as the characteristic path, clustering, degree distributions and resilience to random or targeted failures.

Before discussing random and regular graphs, we introduce two measures which are important for the characterization of network topology: clustering coefficient and characteristic path.

Clustering coefficient is a measure indicating how many nodes tend to cluster together. There are many possible definitions for this coefficient, and we report here two of those which are described in [27]:

- \( C = \frac{3}{n} \sum_{i} C_i \) where \( C_i = \frac{\text{number of triangles connected to vertex } i}{\text{number of triples centered on vertex } i} \).
  
For vertex with degree 0 or 1, we put \( C_i = 0 \).

The characteristic path \( l \) of a graph represents the average path between 2 nodes in the network and can be defined as:

\[
l = \frac{1}{2n(n+1)} \sum_{i \geq j} d_{ij} \tag{1.2}
\]

where \( d_{ij} \) is the shortest distance between vertex \( i \) and vertex \( j \) [27].

Such features are used to discriminate between different network topologies, where with network topology one means the characterization of the network in terms of number of nodes, link distribution among nodes, presence of motifs (sub-networks which are statistically enriched in the graph), and so forth.

Among the different network topologies that there exist, two are of particular interest because they can be seen as basic topologies: the random and the regular ones. Once we will have described these topologies, we will show other two important types of network within the biological context: small-world networks and scale-free networks.

1.2.2 Random and Regular networks

To create a random graph of \( n \) nodes, the simplest model used is the one in which each pair of the \( n \) vertices is connected (or not) with a certain probability \( p \) [28].

The number of edges owned by a node is called node degree. The degree distribution for a network is defined as the empirical distribution of the node degrees. Such distribution can also be seen as the probability \( P(k) \) that a node has exactly \( k \) edges. Since in random graphs the edges are
placed randomly with probability $p$, the degree distribution for such graphs is modeled as a Poisson distribution:

$$P(k) = \frac{e^{-\lambda} \lambda^k}{k!}$$

where $<k> = \lambda$ represents the average node degree for the network.

The clustering coefficient of a random graph is usually low (as for the characteristic path length), and it is defined as:

$$C = p = \frac{<k>}{N}.$$

For what concerns regular networks (also referred to as lattices), these are made up by nodes having the same number of links. The simplest example of regular network is the ring, in which each node is connected to its direct neighbors, both clockwise and anticlockwise. Further generalization can be obtained by adding links and maintaining the network conformation: for example, having each node connected to the $k_1 ... k_s$ nodes on its left and on its right (see Figure 1.9 (b)). These kind of networks are characterized by long characteristic path and high clustering [29].

![Figure 1.9: (a) Random network showing a triangle made up by the triple (0, 1, 3). The connected triples are: (1, 0, 3), (0, 1, 3), (0, 3, 1), (2, 3, 0), (2, 3, 1), (3, 2, 4), (2, 4, 5). This graph has clustering coefficient $C = \frac{3}{7}$ if we consider the first of the two previous definitions. (b) A 1-dimensional lattice (regular network) with 7 nodes: each node is connected with the first and second nodes on its left and on its right.](image)

1.2.3 Small-world and Scale-free networks

A small-world network presents intermediate characteristics between a random and a regular network. It can be built starting from a (1-dimensional)
lattice and adding (or re-placing) edges in order to create a few number of shortcuts linking remote parts of the network [29].

Thanks to the random shortcuts, a small-world network presents a short characteristic path whereas, thanks to its background regular structure, the cluster coefficient remains high (see Figure 1.10).

Formally speaking, *scale-free* networks have a degree distribution that (asymptotically) follows the power law:

\[ P(k) \sim tk^{-\gamma} \]

where \( t \) is a normalization constant, and the exponent \( \gamma \) is usually included in the interval \( 2 < \gamma < 3 \).

The nodes with the highest degree are called hubs (see Figure 1.10), and they are responsible for the network’s predisposition to resist to random failure. Major hubs hold the network together, guaranteeing connectivity among nodes even when non-hubs nodes are randomly removed. If failures occur randomly, given that the vast majority of nodes present small degree, the likelihood that a hub would be affected is almost negligible: therefore, the overall connectivity of the network is secured.

Even if a hub collapses, the network will generally not lose its connectedness, thanks to the remaining hubs. However hubs are a double edged sword: if we choose a few major hubs and force them to collapse, the network turns into a set of rather isolated graphs: the resilience against random failure is balanced by vulnerability towards targeted attacks.

These properties of strength and weakness have been studied using percolation theory and the interested reader can refer to [30]. For more general information, the reader is referred to [31, 27, 32].

![Figure 1.10](image-url)

Figure 1.10: (a) Small-world network. (b) Scale-free network, with hubs 0 and 8.
1.2.4 Network distances

A key point in the network analysis is to glance at the differences between two graphs (in terms of graph-distance). There are a number of possible distances available in literature; here we have chosen the HIM distance (Hamming-Ipsen-Mikhailov), recently defined in [33]. In order to better understand how this distance works, we first show how the Hamming and the Ipsen-Mikhailov distances are defined.

Hamming distance

In coding theory, the Hamming distance is defined as the number of positions where two strings differs from each other. Since the beginning of network distance theory, it has been widely used in order to evaluate differences between two networks [34]. This is done measuring the matchings links within the two adjacency matrices corresponding. In other words, the Hamming distance compute the minimum number of substitutions due to transform the first string (or matrix) in the second one.

Given two networks $N_1$, $N_2$ and their adjacency matrices $A_{ij}^{(1)}$, $A_{ij}^{(2)}$, the Hamming distance is mathematically defined as:

$$H(N_1, N_2) = \frac{1}{N(N-1)} \sum_{1 \leq i \neq j \leq N} |A_{ij}^{(1)} - A_{ij}^{(2)}|$$

As a distance, this quantity is always in the interval $[0,1]$.

Ipsen-Mikhailov distance

Given an undirected network, its Laplacian matrix $L$ is defined as $L = D - A$, where $D$ is the diagonal matrix which has the vertex degrees as elements different from zero, whereas $A$ is the adjacency matrix. Being this $L$ matrix positive and semidefinite, its eigenvalues are $0 = \lambda_0 \leq \lambda_1 \leq \ldots \leq \lambda_{N-1}$. These eigenvalues are closely related to the physical interpretation of the Ipsen-Mikhailov distance [35, 36, 37].

The $N$-nodes network plays the role of a molecule of $N$ atoms connected by identical springs (see Fig. 1.11), [38]. This system is described by $N$ differential equations:

$$\ddot{x}_i + \sum_{1 \leq j \leq N} A_{ij}(x_i - x_j) = 0 \text{ for } i = 0, \ldots, N - 1.$$ 

The frequencies $\omega_i$ of this particular model are related to the previous eigenvalues, being defined as $\lambda_i = \omega_i^2$, with $\omega_0 = 0$. The reader is referred to [39], [40] for a better understanding of the relation between the spectral properties of a network and its dynamical structure.
1.2. NETWORK THEORY

Figure 1.11: A three nodes network as a oscillatory system.

Table 1.3: Adjacency matrices for the weighted directed network (two alternative matrices, with sign indicating direction or asymmetric, with the (positive) value only in entry \((i, j)\) if \(i \rightarrow j\) in Fig. 1.8 and its topology; nodes ordering is clockwise starting from the top node.

<table>
<thead>
<tr>
<th>Network</th>
<th>Adjacency matrix</th>
</tr>
</thead>
</table>
| ![Network](image) | \[
\begin{pmatrix}
0 & 0.33 & 0 & 0 & 0 & 0.5 \\
(-0.33) & 0.12 & 0.85 & 0 & 0 & 0 \\
0 & (-0.85) & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & (-0.25) \\
0 & 0 & 0 & 0 & 0 & (-0.75) \\
(-0.5) & 0 & 0 & 0.25 & 0.75 & 0
\end{pmatrix}
\] |

| ![Network](image) | \[
\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 1 & 0
\end{pmatrix}
\] |
CHAPTER 1. MATHEMATICAL BACKGROUND

Considering the sum of the Lorentz distributions (or Cauchy distributions), it is possible to defined the spectral density for a graph as

$$\rho(\omega, \gamma) = K \sum_{1 \leq i \leq N-1} \frac{\gamma}{(\omega - \omega_i)^2 + \gamma^2}$$

where $\gamma$ is the half-width at half-maximum while K is a normalization constant used in order to obtain $\int_0^\infty \rho(\omega, \gamma) d\omega = 1$.

Given all these parameters, one can define the spectral distance $\tau_\gamma$ between two N-nodes graphs $G$ and $H$ as

$$\tau_\gamma(G, H) = \sqrt{\int_0^\infty [\rho_G(\omega, \gamma) - \rho_H(\omega, \gamma)]^2 d\omega}.$$

Once defined $\epsilon_N$ and $f_N$ as the empty and full networks of N nodes respectively, and defined $\gamma$ as the (unique) solution of $\tau_\gamma(\epsilon_N, f_N) = 1$, it is possible to define the Ipsen-Mikhailov distance as

$$\tau(G, H) = \tau_\gamma(G, H) = \sqrt{\int_0^\infty [\rho_G(\omega, \gamma) - \rho_H(\omega, \gamma)]^2 d\omega}.$$

where, as for the Hamming distance, $\tau(G, H) \in [0, 1]$, [41].

HIM distance

Given the two mathematical definitions of Hamming and Ipsen-Mikhailov distances, the Him distance is defined as a Cartesian product of two metric spaces: the space of all undirected networks of N nodes $\mathcal{N}(N)$ endowed with the Hamming distance and the same space endowed with the Ipsen-Mikhailov spectral distance [33].

Given two graphs $N_1$ and $N_2$, if we denote the Hamming distance with $H(N_1, N_2)$ and the Ipsen-Mikhailov distance with $\tau(N_1, N_2)$, the resulting HIM distance is given by

$$HIM(N_1, N_2) = \frac{\sqrt{2}}{2} \sqrt{H^2(N_1, N_2) + \tau^2(N_1, N_2)}$$

Example of application

Consider the two networks $I_1, I_2 \in \mathcal{N}(8)$ with corresponding adjacency matrices $A^{I_1}, A^{I_2}$ shown in Figures 1.12 and 1.13.
1.2. NETWORK THEORY

\[ A^I_1 = \begin{pmatrix}
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
\end{pmatrix} \]

Figure 1.12: Adjacency matrix and graphical representation of \( I_1 \)

\[ A^I_2 = \begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
\end{pmatrix} \]

Figure 1.13: Adjacency matrix and graphical representation of \( I_2 \)
 CHAPTER 1. MATHEMATICAL BACKGROUND

The Hamming distance between $I_1$ and $I_2$ is

$$ H(I_1, I_2) = \frac{1}{N(N-1)} \sum_{1 \leq i \neq j \leq N} |A_{ij}^1 - A_{ij}^2| $$

$$ = \frac{1}{56} \sum_{1 \leq i \neq j \leq 8} |A_{ij}^1 - A_{ij}^2| $$

$$ = \frac{28}{56} = 0.5. $$

From the spectral point of view, the corresponding Laplacian matrices and eigenvalues are

$$ L_{I_1} = \begin{pmatrix} 3 & -1 & 0 & 0 & -1 & 0 & 0 & -1 \\ -1 & 3 & 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 2 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 2 & -1 & -1 & 0 & 0 \\ -1 & 0 & 0 & -1 & 2 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 2 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & 3 & -1 \\ -1 & -1 & 0 & 0 & 0 & 0 & -1 & 3 \end{pmatrix} $$

$$ \text{spec}(L_{I_1}) = \begin{pmatrix} 0 \\ 0.657077 \\ 2.529317 \\ 4 \\ 4 \\ 4.813607 \end{pmatrix} $$

$$ L_{I_2} = \begin{pmatrix} 3 & -1 & 0 & 0 & 0 & -1 & -1 & 0 \\ -1 & 3 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 2 & 0 & 0 \\ -1 & 0 & 0 & -1 & 0 & 0 & 3 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 & 2 \end{pmatrix} $$

$$ \text{spec}(L_{I_2}) = \begin{pmatrix} 0 \\ 0.340321 \\ 1.145088 \\ 3 \\ 3 \\ 3.854912 \\ 4.659679 \end{pmatrix} $$

From the above spectra, we can compute the corresponding Lorentz distributions $\rho_{I_{1,2}}(\omega, \tau)$, where $\tau = 0.4450034$: their plots are shown in Fig. 1.14.

The resulting Ipsen-Mikhailov distance is
1.2. NETWORK THEORY

\[ \epsilon(I_1, I_2) = \sqrt{\int_0^\infty \left[ \rho_{I_1}(\omega, \gamma) - \rho_{I_2}(\omega, \gamma) \right]^2 d\omega} = 0.1004144 , \]

so that the HIM distance results

\[ \text{HIM}(I_1, I_2) = \frac{\sqrt{2}}{2} \| (H(I_1, I_2), \epsilon(I_1, I_2)) \|_2 \approx 0.707168 \sqrt{0.5^2 + 0.1004144^2} \approx 0.3606127 . \]

The situation can be graphically represented as in Fig. 1.15: the two networks are quite different in terms of matching links, but their structures are not so far away.

Figure 1.14: Lorentzian distribution of the Laplacian spectra for \( I_1 \) and \( I_2 \). Vertical lines indicate eigenvalues.
Figure 1.15: HIM($I_1, I_2$) in the Hamming/Ipsen-Mikhailov space
Chapter 2

Network inference background

2.1 Networks in Cell Biology

A cell is a vast complex reality constituted by thousand biological components which interact performing biochemical reactions necessary for life. According to a bio-mathematical hypothesis, the network hypothesis, a cell is describable as a set of interconnections among its component molecules. From this point of view, it becomes convenient to describe the functioning of the cell focusing on these particular interactions rather than on the molecules themselves [42].

Introduced in 1958 by F. Crick [43], the central dogma in molecular biology describes the way in which information flows within the cell; it then defines relations among DNA, RNA and proteins. The first important passage is called replication: DNA shares information creating a copy of itself. As second step, DNA passes the message to RNA (through transcription), arriving in this way to the third step, the translation, in which RNA transfer it to proteins.

This highly regulated process, the gene transcription, is the scenario where other important biological entities act: the transcription factors (TFs). They are proteins able to bind to specific regions (promoter regions) of the gene and to control the process of transcription. Being proteins, they are gene products themselves; this imply the presence of a backward flow of information from proteins to DNA in the form of a gene-to-protein-to-gene control. As an example, there exist genes which encode proteins that bind to TFs in order to inactivating them (gene-to-protein-to-protein-to-gene control). Other complications come from the not complete cross-specificity between TFs and promoter regions: the same TF can bind to different promoters area and vice versa. These are just few reasons of the complexity of this process.
However, the comprehension of this complex mechanism of regulation is rapidly increasing; it is nowadays a common thought that, in order to comprehend the deep manner in which a cell acts, is required a greater understanding of how the identity of the cell can be described by its sets of interactions.

It is therefore of great importance the investigation of the way in which proteins (or genes) interact among themselves (or with other molecules). Networks provide a suitable way to deal with this problem [44]. They represent a powerful method in order to investigate the data in one single object, and to quantitatively assess the fragility and robustness of the biological system. After having established the topology of the network, it must be studied in its changes with time. These two aspects are at the heart of the present research about biological networks. We show here a few kind of application of these structures on the study of these concepts.

**Protein-protein interaction networks**

A protein-protein interaction network represents different kind of relations (edges) among proteins (vertices). A variety of different techniques have been developed in order to recognize the disparate nature of these interactions [45].

**Metabolic networks**

The term *metabolism* refers to the entire chemical process, in action within living organisms, that create molecules necessary to the cellular functions, through the transformation of the resources taken from the environment. It can be seen as a linear sequence of chemical transformations The *metabolites* are the chemical compounds acting in a reaction, while *enzymes* are the specific proteins which catalyzed the reactions. In these kind of networks, vertices represent metabolites, whereas edges connect vertices participating in the same reaction.

**Signaling networks**

Cells are able to perceive changes in their environments and react to them in an appropriate way. This sensitivity is achieved by means of many cell surface-located sensor proteins; they can modify their behavior under the stimuli coming from changes in, temperature, pH or salt concentrations. The modification of their behavior create signals that, after having reached the cytoplasm, induce the activation of genes encoding proteins needed for the cellular response. These are just a few examples of how important is the modeling of network structures in the study of the cell functionality. The network approach
to cell biology aims to understand the intertwined structures of the whole system within a unified framework.

2.2 State-of-the-art

Biological network inference is the process of inferring and predicting notions about biological networks. This process is applied to high-throughput data and can provide important information about the gene expression in cells. The results of network inference are different depending on used tools and can be highly complementary. This complementarity leads to the idea that a certain tool is more appropriate than others for a specific research question.

The era of system biology began with the idea that genes and proteins work together in intricate networks. Understanding the gene co-expression network and the transcription-regulatory network (TRN, for short) is therefore crucial in order to comprehend the cellular behavior (see Figure 2.1 from [46]). The number of computational methods which are developed (and that are being developed nowadays) to reconstruct TRNs from genomewide expression data is increasing more and more; they are grouped under the name of methods expression-centered methods. Among them, there are the Module inference methods, which focus on the co-expression network, and rely on the guilt-by-association principle to identify functional relationships between genes. On the other hand, methods that infer TRNs go one step beyond; they are able to infer causality relationships (enhancer or inhibitory) in the network by the identification of the transcriptional programmes of the genes, describing how transcription factors (TFs) cause the observed changes in their target genes expression levels (2.1, reported from [46]).

We provide a scheme (reported in Figure 2.2 from [46]) that shows the state-of-the-art transcriptional inference methods, classified on the basis of the strategies used to solve the inference problem.

Huge amount of data will create the problem of complexity; the larger the dataset is, the more difficult it will be to find the unique solution that best interprets the biological truth. To tackle this problem, network inference methods adopt different strategies with the aim of reducing the search (see 2.2 from [46]).

The most used strategy in order to do so, is the 'Conceptualization by simplifying biological reality': TRNs have been shown to have a modular structure, that is a structure of overlapping modules of functionally related genes. Genes belonging to the same module act in concert, explaining their coordinated expression behavior. Modules are then identified by methods that rely on clustering (common technique for statistical data analysis). These kind of network inference procedures assign a regulatory programme to each of these modules, instead of assigning an individual programme to each single gene, as generic direct network inference methods do. This pro-
procedure lowers the number of interactions, simplifying the evaluation during the inference process.

A second strategy focuses on the extension of the expression data with other available information. Integrative methods are able to combine the expression data with complementary data describing the TRN from a different point of view, such as motif data; these methods often give in output a more complete picture of the biological network.

The third strategy mentioned is based on query-driven methods, which reduce the search space by restricting the research field. Instead of looking for a global pattern, as global inference methods do, these methods concentrate their power on a smaller set of core genes (subnetwork of interest), and then they expand on the entire network based on this subnetwork.

Supervised (and semisupervised) methods are considered the fourth strategy: they treat the network inference problem as a classification problem, valorizing the available information in a different manner.

Given that each strategy acts on different assumptions and works with different constraints, a specific strategy or combination of strategies will determine a specific type of interactions (modeled with links). Obtaining the same inferred results from different methods can be very difficult, as illustrated in 2.3 reported from [46]. This means that the discrepancy in the prediction of the interactions is not due to just the failure of one of the methods, but it is rather the complex characteristic of complementarity of the different methods. For the moment, no single best method exists; different methods highlight different kind of interaction. These are the reasons which lead to the idea of combinations of complementary methods, offering in this way an improving in the breadth and accuracy of the predictions themselves.

Finding the optimal solution is obviously a non-trivial problem because of the large search space. For methods that can lead to different possible solutions, a global solution allows the accuracy of the predicted interactions to be increased by better approximating the global solution. State-of-the-art inference tools are based on unique combinations of strategies. They each have different strengths and limitations and highlight complementary aspects of the network. Module-based inference methods are to be preferred when using an extensive data set, instead of applying direct inference methods. When the aim is to recreate the complete TRN, a global approach could be more suitable than a query-driven one.

2.3 Correlation & DTW-based new method

In modern computational biology, substantial efforts are put into inferring the structure of biological networks starting from measurable biological data. Although many improvements have been made, several issues still remain
Figure 2.1: (taken from [46])

(a) **Co-expression network**: this is a network representation in which the nodes represent the genes and the edges represent the degree of similarity in the expression profiles of the genes. Cliques or highly connected subgraphs correspond to modules of co-expressed genes. The edges are undirected, indicating that they represent only a correlation or dependency relationship between the nodes and do not reveal the cause of the relationship.

(b) **Transcription-regulatory network**: this is a bipartite graphical network representation in which the nodes represent either transcription factors (TFs) or target genes (or modules) (see the figure, part b). Edges are directed, as they reflect a causal relationship: they indicate that an observed correlation in the expression patterns of two nodes is caused by a node corresponding to a TF regulating a node corresponding to a target gene. A transcriptional programme corresponds to a set of TFs sharing the same set of target genes, ideally under a similar subset of conditions.
CHAPTER 2. NETWORK INFERENCE BACKGROUND

Figure 2.2: (taken from [46]) Module inference methods search for sets of co-expressed genes. The major goal of network inference (NI) methods, on the other hand, is to search for a regulatory programme that explains an observed expression behavior. NI methods can be categorized according to the strategies that they use to cope with the problem of underdetermination. Direct NI methods consider all genes on an individual basis, whereas module-based NI methods conceptualize the network by treating sets of co-expressed genes as single entities (modules). NI and module interference methods can be further divided according to whether they complement expression data with additional data sources (integrative methods) or use expression data only (non-integrative methods). Supervised and semi-supervised methods treat the inference problem as a classification problem, whereas unsupervised methods do not. The output of the methods can be global, indicating that they search for global patterns in the data, or query-driven, starting from a predefined set of core genes or core pathways and expanding on those. Most of the available programs can be used in either a query-driven or a global mode. The methods indicated in pink are specifically designed to be query driven. CLR, context likelihood of relatedness; COALESCE, combinatorial algorithm for expression- and sequence-based cluster extraction; DISTILLER, data integration system to identify links in expression regulation; GPS, gene promoter scan; LeMoNe, learning module networks; SEREND, semi-supervised regulatory-network discoverer; SIRENE, supervised inference of regulatory networks.
Various network inference methods were run on the same Escherichia coli gene expression compendium and their results were compared. The proportion of shared predictions out of the total number of predictions ranges from 5.7% to, at most, 24%. The overlap with RegulonDB (number of interactions in common with the external standard / total number of predicted interactions) ranges from 15% to 18%, and the overlap with chromatin immunoprecipitation-on-chip (ChIP-chip) data ranges from 2% to 3%, with a very low performance for CLR (context likelihood of relatedness) predictions compared with ChIP-chip data (1%).

(a) A mutual comparison between the results of the module-based approach Stochastic LeMoNe (learning module networks) and the direct method CLR, both of which are non-integrative and unsupervised, using the known network data in RegulonDB as an external standard. (b) A comparison between the results obtained using CLR and the supervised method SIRENE (supervised inference of regulatory networks; both methods are non-integrative and direct). Available ChIP-chip data for several E. coli regulators was used as an external validation standard, as SIRENE uses the information in RegulonDB to make its predictions. (c) A comparison between the results of the non-integrative method SIRENE and the integrative method SEREND (semi-supervised regulatory-network discoverer), which combines expression data with motif data (both methods are supervised and direct). Available ChIP-chip data was used as an external standard, as in part b.
Within this thesis, we have investigated well-known methodologies for reverse engineering of regulatory networks, and introduced a new method in order to overcome some of the existing limitations.

Gene networks can be modeled and simulated using various approaches, among which correlation, linearized differential equations, boolean networks and many other (see [48], [49]). Regarding to the reverse engineering methods, we focused both on correlation, DTW, and a newly introduced combination of them.

The basic idea, is to use these two form of distance to decide whether two genes (described by their time series coming from Microarray analysis) are linked or not. Briefly speaking, gene chips, or microarrays, are new technologies for large-scale gene expression monitoring, and they are used to detect differences in mRNA expression levels of thousands of gene at one time. From this huge amount of data, only few time series will be considered, using machine learning pipelines in order to discriminate important genes from the others (see [14], [50]).

2.3.1 Building the optimal correlation and DTW matrices

The first step in order to build up optimal correlation (or DTW) matrices is the (n x p) matrix of the time series (TS), defined as TSM (Time Series Matrix); the \( k \)th row of this matrix represent the time series associated to the \( k \)th gene. Then we will create the weighed correlation matrix \( C \) (in which the element at position \((i,j)\) is given by the Pearson’s correlation between TS’s \(i\) and \(j\)) and the weighed dtw matrix \( D \) (in which the element at position \((i,j)\) is given by the dtw distance between TS’s \(i\) and \(j\)). The next step is to choose a set of fixed quantiles (from 0 to 1, by steps of 0.005) that are used to binarized the matrices \( C \) and \( D \). We evaluate the thresholds of both reversed engineering methods (i.e., based on correlation or dtw) making use of quantiles: among all distances (first correlations and after for the dtw distances) we will calculate the quantiles and we will use these as thresholds for the binarizing step. We create different binarized solutions \( C^b_q \) and \( D^b_q \) as follows:

1. all corresponding quantiles for both correlations and dtw distances are calculated;
2. the \((i,j)\) element of \( C^b_q \) is set to 1, if the corresponding correlation (in absolute value) is greater than or equal to its corresponding quantile, to 0 otherwise;
3. the \((i,j)\) element of \( D^b_q \) is set to 1, if the corresponding dtw distance is lower than or equal to its corresponding quantile, to 0 otherwise.

Thus, for each quantile we have two binarized matrices \( C^b_q \) and \( D^b_q \). Finally, according to the flag parameter ‘dist.flag’ (0 = HIM distance, 1 =
2.3. CORRELATION & DTW-BASED NEW METHOD

zoom distance (not described here), we compute the HIM distances from Ad (initial adjacency matrix) to each $C^b_q$ and $D^b_q$. The optimal quantile thresholds are finally selected for $C^b$ and $D^b$.

2.3.2 New combining method

Following biological considerations, we decided to combine both methods (correlation and dtw) in the following way (see Figure 2.4 for a better comprehension):

1. dtw is used for finding the optimal shift between two given TS’s $i$ and $j$ (see Section 1.1.3 in Chapter 1);

2. subsequently, we create a phantom series representing the second series shifted (back or forward) of the corresponding shift previously calculated;

3. finally, we calculate the Pearson’s correlation between the first series and the phantom one. This correlation will be the CD$[i,j]$ element of the new CD matrix. Given that with this procedure CD$[i,j]$ could be different from CD$[j,i]$, and considering that the final matrix CD has to be symmetric (before being binarized), we decided to assign at both CD$[i,j]$ and CD$[j,i]$ the maximum between them.

2.3.3 Graphs visualization

The optimal binary matrices can be represented graphically and compared to the original Ad model. To this goal, we make use of the igraph library (R package), and more particularly of the plot.igraph function.

Node size in the graph representation is coded according to the node degree (i.e., higher degree is reflected in a larger node size). One could consider to code other topological parameter as well, such as node centrality or clustering, in order to better visualize similarities and differences between graphs.
Figure 2.4: Schematic diagram of the combined method. (a) Starting from a set of time series (one per gene), for each possible pair of time series (b) we make use of the DTW to estimate the relative time shift (e.g., of the black time series with respect to the blue one). Next, we create a new time series based on the estimated shift (c) (in red) and finally evaluate the Pearson correlation between the newly created time series and the blue one. Such correlation will be used to identify links within the gene regulatory network (d). To conclude, the HIM distance is used to evaluate the inferred network (e).
Chapter 3

Results on general purpose synthetic data

Synthetic data are very important in order to train a new model. In this chapter we introduce the approach we have followed to create our synthetic data. Furthermore, we describe our experimental design and present the results obtained applying our model to the synthetic data.

3.1 Synthetic data

In this section, we present the procedure we followed to create the synthetic data. We will show the approach adopted to create biological synthetic data and we will use the method to infer regulatory networks from them (described in Chapter 2). We will show how we have created the initial adjacency matrix and the simulated time series. After that, we will present our method to simulate co-regulated genes, followed by the experiments and their results.

3.1.1 Adjacency matrix and simulated biological time series

As a first step, we create the \((n \times n)\) initial adjacency matrix \(A_d\) as having either a scale-free, small-world or random topology. This is achieved by using the available generating functions from the igraph library, such as the \(barabasi.game\) function (see the parameter \(Net\_topology\) in Table 3.1.).

Secondly, after having fixed a length \(p\) (see \(ts\_length\) in Table 3.1) for our time series (TS’s), we generate a first \((n \times p)\) time series matrix (TSM) choosing randomly sub-series (of length \(p\)) from the aami3a TS included in the dtw R package. Thus TSM is the matrix in which the \(i^{th}\) row corresponds to the \(i^{th}\) gene (see Chapter 1). After that, we modify each gene time-course, introducing three kinds of large noise:
CHAPTER 3. RESULTS ON GENERAL PURPOSE SYNTHETIC DATA

• reshuffling: we randomly change the first $x\%$ of the time points, by randomly choosing other time points from the time series. This allows us to uncorrelate the new TS in its initial part. The $x\%$ is chosen randomly from a normal distribution of mean $\beta$ and standard deviation $\beta_{sd}$.

• noise deformation: change all values of TS by randomly adding or subtracting the given percentage of the current TS value (par: $\alpha'$).

• non-linearity deformation: we apply a non-linear transformation, set with the flag parameter $\delta = 0$ ($0 = (TS)^2$, $1 = \log(TS)$, $2 = 1/(TS)$). Such non-linearity is applied to only the first $y\%$ of the TS time points, where $y\%$ is chosen from a normal distribution of mean $\gamma$ and standard deviation $\gamma_{sd}$.

All parameters are shown in Table 3.1. We have checked that the final TS’s are highly uncorrelated by plotting the histograms of all possible correlation pairs and investigating the distribution of such correlations.

3.1.2 Simulation of co-regulated genes

In order to simulate similar expression profiles according to our Ad model, we modify the TSM as follows:

• weighing: every gene’s time course (row of TSM, here indicated by $ts_i$) is replaced by a weighed sum of all the other genes connected to it. For example, if in Ad the gene 1 is linked to genes 2, 3, and 5, then its time-course would be replaced by $ts_1 \leftarrow ts_2 + ts_3 + ts_5 + ts_1$.

Hence, even its own time-course is used in the averaging.

• noise deformation: change all values of $ts_i$ by randomly adding or subtracting the given percentage of the current $ts_i$ value (par: $\eta'$). This is done for all time courses.

• shift: the $ts_i$ is shifted forward for an $g\%$ of the $ts_i$ time points, where $g\%$ is chosen randomly from a normal distribution of mean $\mu$ and standard deviation $\mu_{sd}$. Obviously, doing so one looses the final $g\%$ of the $ts_i$, while the initial $g\%$ is replaced with noise (par: $\theta'$) around the mean of the initial part of the $ts_i$. This is done for all time series.

All these parameters are also shown in Table 3.1.
3.2 Experimental design and results

We present here the results of our experiments. In order to test the effects of the parameters on the final models, we considered 7 different parameter sets. For each of these, we will show two set of results:

- single experiment (N = 1): a plot of HIM distances against quantiles (for correlation, dtw, and combined methods) and the final plot of the graphs.
- repeated experiments (N = 20): table of means and standard deviation, and corresponding boxplot.

Parameter Set 0

As a first scenario, we considered the case of uncorrelated time series. The parameters are shown in Table 3.2. The main goal for this set-up was to investigate how the distances between the reconstructed graphs and the original one are distributed when the reconstruction is based on uninformative data.

Results for the N=20 experiments are shown in Table 3.3 and Figure 3.4. Results for a single experiment (graphs and quantiles) are shown in Figures 3.1 and 3.4.

Parameter Set 1

Referring to Table 3.2, we have introduced the Set 1 in order to test the reconstruction methods under the hypothesis of correlated time series, without noise (see section 3.1.2).

Results for the N=20 experiments are shown in Table 3.3 and Figure 3.5. Results for a single experiment (graphs and quantiles) are shown in Figures 3.2 and 3.5.

The method based on only the dtw does not allow a good reconstruction of the graph: this is to be expected since we haven’t introduced yet shifts. Dtw can not control changes in the y direction properly.

Parameter Set 2

In order to assess the reconstruction performances of the three methods in presence of noise, we have considered the parameter set 2 (see Table 3.2). Time series were created as for parameter set 1, and then modified by both shifting and adding noise.
CHAPTER 3. RESULTS ON GENERAL PURPOSE SYNTHETIC DATA

Results for the N=20 experiments are shown in Table 3.3 and Figure 3.6. Results for a single experiment (graphs and quantiles) are shown in Figures 3.3 and 3.6.

The analysis shows that correlation can not handle shifts properly; the combination of dtw and correlation, on the other hand, improves the results considerably.

Parameter Sets 3 and 4

Parameter sets 3 and 4 were used to investigate the performances of the reconstruction methods when an increasing amount of shifting is considered. Values are reported in Table 3.2.

Results for the N=20 experiments are shown in Table 3.3 and Figures 3.7 and 3.8. Results for a single experiment (graphs and quantiles) are shown in Figures 3.4, 3.5, 3.7 and 3.8.

As expected, the correlation-based method gets worse with increasing amounts of shifting. The best method remains the combination of dtw and correlation.

Parameter Set 5

Parameter set 5 was created to investigate what happens with increasing noise at each single point of the series (see Table 3.2). Time series were created as for parameter set 3 increasing the percentage of applied noise.

Results for the N=20 experiments are shown in Table 3.3 and Figure 3.9. Results for a single experiment (graphs and quantiles) are shown in Figures 3.6 and 3.9.

The best method is still the combination of both correlation and dtw via the phantom series.

Parameter Set 6

Eventually, parameter set 6 was created to investigate what happens with an increase of noise around the mean at the beginning of the shifted series (see section 3.1.2 and Table 3.2). Time series were created as for parameter set 3, increasing the percentage of noise applied before the shifts.

Results for the N=20 experiments are shown in Table 3.3 and Figure 3.10. Results for a single experiment (graphs and quantiles) are shown in Figures 3.7 and 3.10.

The best method, as for parameter set 5, is still the combination of both correlation and dtw via the phantom series.
Table 3.1: Description of the most important parameters used in the script.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net topology</td>
<td>0 = Random Ad, 1 = small-world Ad, 2 = scale-free Ad</td>
</tr>
<tr>
<td>Sparsity Index</td>
<td>sparsity of the initial matrix Ad (only for random topology)</td>
</tr>
<tr>
<td>N. nodes</td>
<td>number of nodes (genes)</td>
</tr>
<tr>
<td>N. iterations</td>
<td>number of iterations of the single experiment</td>
</tr>
<tr>
<td><strong>Large noise to make dissimilar two given TS:</strong></td>
<td></td>
</tr>
<tr>
<td>$\alpha$</td>
<td>% of noise in values of a TS</td>
</tr>
<tr>
<td>$\beta$</td>
<td>mean of % of noise added at the beginning of a TS</td>
</tr>
<tr>
<td>$\beta_{sd}$</td>
<td>sd of % of noise added at the beginning of a TS</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>mean of % of deformation improved</td>
</tr>
<tr>
<td>$\gamma_{sd}$</td>
<td>sd of % of deformation improved</td>
</tr>
<tr>
<td>$\delta$</td>
<td>non linearity: $0 = ^2$, $1 = \log_{10}$, $2 = 1/x$</td>
</tr>
<tr>
<td><strong>Noise to make similarities:</strong></td>
<td></td>
</tr>
<tr>
<td>weighing</td>
<td>0 = no weighing, 1 = weighing (See section 3.1.2)</td>
</tr>
<tr>
<td>$\eta$</td>
<td>% of noise used in make_series_similar_new</td>
</tr>
<tr>
<td>$\mu$</td>
<td>% of shifting used by modifier</td>
</tr>
<tr>
<td>$\mu_{sd}$</td>
<td>sd of the $\mu$</td>
</tr>
<tr>
<td>$\theta$</td>
<td>% of noise around the mean ...used by the modifier</td>
</tr>
</tbody>
</table>
Table 3.2: Most important parameters used in the script. When applicable, mean ($\pm$ sd) is reported; otherwise, only the mean.

<table>
<thead>
<tr>
<th></th>
<th>Set 0</th>
<th>Set 1</th>
<th>Set 2</th>
<th>Set 3</th>
<th>Set 4</th>
<th>Set 5</th>
<th>Set 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>ts_length</td>
<td>72</td>
<td>72</td>
<td>72</td>
<td>72</td>
<td>72</td>
<td>72</td>
<td>72</td>
</tr>
<tr>
<td>Sparsity Index</td>
<td>10%</td>
<td>10%</td>
<td>10%</td>
<td>10%</td>
<td>10%</td>
<td>10%</td>
<td>10%</td>
</tr>
<tr>
<td>N. nodes</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Net topology</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>N. iterations</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Large noise</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\alpha$ (±$\beta_{sd}$)</td>
<td>± 30%</td>
<td>± 30%</td>
<td>± 30%</td>
<td>± 30%</td>
<td>± 30%</td>
<td>± 30%</td>
<td>± 30%</td>
</tr>
<tr>
<td>$\beta$ (±$\beta_{sd}$)</td>
<td>20(± 5)</td>
<td>20(± 5)</td>
<td>20(± 5)</td>
<td>20(± 5)</td>
<td>20(± 5)</td>
<td>20(± 5)</td>
<td>20(± 5)</td>
</tr>
<tr>
<td>$\gamma$ (±$\gamma_{sd}$)</td>
<td>30(± 5)</td>
<td>30(± 5)</td>
<td>30(± 5)</td>
<td>30(± 5)</td>
<td>30(± 5)</td>
<td>30(± 5)</td>
<td>30(± 5)</td>
</tr>
<tr>
<td>$\delta$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Noise to make similarities:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>weighing</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$\eta$</td>
<td>0 %</td>
<td>0 %</td>
<td>10 %</td>
<td>10 %</td>
<td>10 %</td>
<td>20 %</td>
<td>10 %</td>
</tr>
<tr>
<td>$\mu$</td>
<td>0 %</td>
<td>0 %</td>
<td>5 %</td>
<td>10 %</td>
<td>20 %</td>
<td>10 %</td>
<td>10 %</td>
</tr>
<tr>
<td>$\mu_{sd}$</td>
<td>0 %</td>
<td>0 %</td>
<td>1 %</td>
<td>3 %</td>
<td>5 %</td>
<td>3 %</td>
<td>3 %</td>
</tr>
<tr>
<td>$\theta$</td>
<td>0 %</td>
<td>0 %</td>
<td>5 %</td>
<td>5 %</td>
<td>5 %</td>
<td>5 %</td>
<td>10 %</td>
</tr>
</tbody>
</table>
3.2. EXPERIMENTAL DESIGN AND RESULTS

Table 3.3: Means and Sd’s of minimum HIM distances for different reverse engineering methods, calculated over N = 20 experiments. (e.g. the fifth row show us mean and sd (over the 20 experiments) of the minimum with the method of shift + correlation). For each method and parameter set, we show both the best distance and the corresponding quantile threshold.

<table>
<thead>
<tr>
<th>Method</th>
<th>Cor</th>
<th>CorQ</th>
<th>Dtw</th>
<th>DtwQ</th>
<th>Cor_Dtw</th>
<th>Cor_DtwQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set 0</td>
<td>0.248(0.02)</td>
<td>0.871(0.03)</td>
<td>0.246(0.03)</td>
<td>0.225(0.05)</td>
<td>0.251(0.02)</td>
<td>0.774(0.02)</td>
</tr>
<tr>
<td>Set 1</td>
<td>0.062(0.02)</td>
<td>0.836(0.02)</td>
<td>0.185(0.03)</td>
<td>0.244(0.03)</td>
<td>0.064(0.03)</td>
<td>0.736(0.02)</td>
</tr>
<tr>
<td>Set 2</td>
<td>0.152(0.04)</td>
<td>0.852(0.02)</td>
<td>0.167(0.04)</td>
<td>0.245(0.02)</td>
<td>0.114(0.04)</td>
<td>0.740(0.04)</td>
</tr>
<tr>
<td>Set 3</td>
<td>0.196(0.04)</td>
<td>0.860(0.02)</td>
<td>0.200(0.04)</td>
<td>0.253(0.04)</td>
<td>0.131(0.03)</td>
<td>0.754(0.02)</td>
</tr>
<tr>
<td>Set 4</td>
<td>0.209(0.03)</td>
<td>0.851(0.04)</td>
<td>0.177(0.03)</td>
<td>0.251(0.04)</td>
<td>0.149(0.06)</td>
<td>0.755(0.03)</td>
</tr>
<tr>
<td>Set 5</td>
<td>0.192(0.04)</td>
<td>0.838(0.04)</td>
<td>0.209(0.04)</td>
<td>0.243(0.04)</td>
<td>0.161(0.03)</td>
<td>0.746(0.04)</td>
</tr>
<tr>
<td>Set 6</td>
<td>0.188(0.03)</td>
<td>0.846(0.02)</td>
<td>0.170(0.04)</td>
<td>0.255(0.05)</td>
<td>0.143(0.04)</td>
<td>0.744(0.03)</td>
</tr>
</tbody>
</table>
CHAPTER 3. RESULTS ON GENERAL PURPOSE SYNTHETIC DATA

Table 3.4: (a) Boxplot of the results related to parameter set 0. (b) Black line: correlation method. Red line: DTW method. Green line: Combined method.

Figure 3.1: Reverse engineered graphs with corresponding optimal distances for parameter Set 0. (a) Correlation method. (b) DTW method. (c) Correlation after DTW method. (d) Initial network.
3.2. EXPERIMENTAL DESIGN AND RESULTS

Table 3.5: (a) Boxplot of the results related to parameter set 1. (b) Black line: correlation method. Red line: DTW method. Green line: Combined method.

Figure 3.2: Reverse engineered graphs with corresponding optimal distances for parameter Set 1. (a) Correlation method. (b) DTW method. (c) Correlation after DTW method. (d) Initial network.
Table 3.6: (a) Boxplot of the results related to parameter set 2. (b) Black line: correlation method. Red line: DTW method. Green line: Combined method.

Figure 3.3: Reverse engineered graphs with corresponding optimal distances for parameter Set 2. (a) Correlation method. (b) DTW method. (c) Correlation after DTW method. (d) Initial network.
3.2. EXPERIMENTAL DESIGN AND RESULTS

Table 3.7: (a) Boxplot of the results related to parameter set 3. (b) Black line: correlation method. Red line: DTW method. Green line: Combined method.

Figure 3.4: Reverse engineered graphs with corresponding optimal distances for parameter Set 3. (a) Correlation method. (b) DTW method. (c) Correlation after DTW method. (d) Initial network.
Table 3.8: (a) Boxplot of the results related to parameter set 4. (b) Black line: correlation method. Red line: DTW method. Green line: Combined method.

Figure 3.5: Reverse engineered graphs with corresponding optimal distances for parameter Set 4. (a) Correlation method. (b) DTW method. (c) Correlation after DTW method. (d) Initial network.
3.2. EXPERIMENTAL DESIGN AND RESULTS

Table 3.9: (a) Boxplot of the results related to parameter set 5. (b) Black line: correlation method. Red line: DTW method. Green line: Combined method.

Figure 3.6: Reverse engineered graphs with corresponding optimal distances for parameter Set 5. (a) Correlation method. (b) DTW method. (c) Correlation after DTW method. (d) Initial network.
Table 3.10: (a) Boxplot of the results related to parameter set 6. (b) Black line: correlation method. Red line: DTW method. Green line: Combined method.

Figure 3.7: Reverse engineered graphs with corresponding optimal distances for parameter Set 6. (a) Correlation method. (b) DTW method. (c) Correlation after DTW method. (d) Initial network.
3.3 Discussion

In this chapter we have presented different reverse engineering methods to reconstruct regulatory networks. Synthetic data reflecting biological genic profiles were used as benchmark. In order to do so, we have built a simulator pipeline that creates time series (to be interpreted as genic profiles), making use of biologically realistic shifts (simulating delays) and noise. We have then computed correlations and Dtw distances as basic ways to reconstruct the initial network. The novelty which has been introduced is the combination method, which uses the Dtw to recognize the best shift between any pair of time courses, coupled with the correlation method. Such approach came up to be the best method among the three, so in the next chapter we will apply it to biologically-like synthetic data, coming from an open-source tool for in silico benchmark generation.
In the previous chapter we have made use of simulated genetic profiles as benchmark for our new inferring method. Similarities between genetic profiles were introduced depending on the initial adjacency matrix Ad. Subsequently, we have applied our inferring methods to the time courses obtained with our simulation. Our results showed that the combined method based on both correlation and DTW outperformed the methods based only on correlation or on DTW. However, the genetic profiles created with our algorithms could only reproduce simple biological properties. Different tools are available to generate more biologically realistic genetic profiles: in this chapter we will introduce one of these tools and used it to generate new ground-truth genetic profiles in order to assess the performances of our inferring method.

4.1 GeneNetWeaver (GNW)

The aim of this section is to give to the reader a fairly complete overview on the open-source tool GeneNetWeaver. We will explain how it works and how we have used it in order to create more biologically realistic synthetic data.

4.1.1 The GNW tool for simulation of regulatory networks

GeneNetWeaver 3.1 Beta (GNW) is an open-source tool for in silico benchmark generation and performance profiling of network inference methods (see [51] and [52]). It has been used to provide data for an annual network inference challenge, the DREAM project (Dialogue for Reverse Engineering Assessments and Methods, [53]).

GNW provides a databases of well-characterized real biological networks (e.g., Ecoli, Yeast, etc.); for each of these organisms, several subnetwork
of the real regulatory networks are available (e.g., 10 nodes, 100 nodes, etc.). Moreover, networks used in previous editions of the DREAM challenge are also available. To our purpose, we have chosen the DREAM3_In-Silico_Size_10, and in particular the the model InSilicoSize10-Ecoli1 (hence, a network made of 10 nodes, in order to be consistent with the experiments shown in the previous chapter).

Once the initial adjacency matrix has been chosen, the tool builds first a realistic kinetic model for the network. It is then possible to create benchmarks in two distinct ways: using a deterministic model (ODE’s) or a stochastic model (SDE’s). It is important to stress that the biological background of these dynamical network models include important factors such as the difference between regulators and regulated genes, with realistic simulation of enhancing and/or inhibiting interactions.

The algorithm works by perturbing the steady-state of the network and sampling the expression level of the different genes at different time points. The length of acquired time series and the time resolutions can be defined by the user. Similarly, the number of perturbations can also be defined. At each perturbation, a new set of time series is generated, one for each gene.

4.1.2 Synthetic data generation

We have chosen to simulate data for the following conditions:

- network: DREAM3_In-Silico_Size_10 (InSilicoSize10-Ecoli1)
- model: deterministic (ODE)
- platform being simulated: microarray with noise
- number of perturbations: 30

The number of perturbations was set to 30 to be sure we have sufficient information regarding all interactions in the network (see subsection 4.2.1).

4.2 Improved algorithms for inferring regulatory networks

In this section, we want to explain how we have improved the application of the phantom (combined) method in order to better infer gene regulatory networks from the benchmark obtained by the tool GNW. This will be done in subsections 4.2.2 and 4.2.3. As said before, we prefer to explain why we have chosen to create 30 time series and we will do this in the next section.
4.2. IMPROVED ALGORITHMS FOR INFERRING REGULATORY NETWORKS

4.2.1 Issues with perturbation-based simulated data

In order to generate a time series related to a dynamical network, what GNW does can be basically divided into two shares:

- excitation: only few transcription factors (TF’s) among all are perturbed (i.e. activated in order to vivify the model interactions) through the deterministic model
- recording: a time series for each gene of the network is recorded from the moment of the excitation on (until the length previously decided by the user)

This procedure leads to a problem (for our purpose) within the entire set of time series produced: given that only few TF’s are perturbed, then with high probability not all regulated genes will be affected by the perturbations, and so we have thought that this would probably cause problems to correlation and DTW. For instance, if a gene in the network is regulated only by a TF which is not perturbed by the model in the creation of the time series, this gene would be difficult seen related to that TF. Thus our inferred network would have problems with this link.

This is the problem that leads us to the decision of generate 30 time series sets, each one composed by 10 time series (of length \( l = 100 \)). In this way, for each set the model will excite different TF’s, allowing us to have a higher probability of seeing every link existing (enhancer or inhibitor) in the network. Once discussed this problem, we can proceed to the two different modification of the procedure viewed in the Chapter 3 in order to infer networks from this benchmark.

4.2.2 First method

For the first method of analysis, we start from the 30 time series set described before. Now we explain what is the procedure followed for each of them.

As initial step, the reader is referred to the Chapter 3 and in particular to the table 3.3. From this table, we extract the mean of each column referred to quantiles (CorQ, DtwQ, Cor_DtwQ) in order to use the mean best quantile coming from experiments on our former synthetic data. Doing so, we obtain the three quantiles \( Q_c = 0.851 \), \( Q_d = 0.245 \) and \( Q_{combined} = 0.750 \) respectively. Thus, with these particular quantiles, we create the C, D, CD matrices exactly in the same way described in the Chapter 3. Doing this for each time series set, we obtain a set of 90 binarized matrices \( C_k, D_k, CD_k \) where \( k = 1, ..., 30 \). Then, with the aim of investigate which links are the real existing links in the initial network, we want to understand how many times each of them appeared in the 30 sets.
CHAPTER 4. RESULTS ON BIOLOGICALLY-LIKE SYNTHETIC DATA

For this reason, we sum the $C_k$, $D_k$, $CD_k$ over the $k$, obtaining 3 matrices $C_{Tot}$, $D_{Tot}$, $CD_{Tot}$. The generic element $C_{Tot}[i, j]$ is a natural number $0 \leq C_{Tot}[i, j] \leq 30$ indicating how many times the link between gene $i$ and gene $j$ appeared. Now, for each $k = 1, \ldots, 30$ we visualize these 3 total matrices and we binarized them putting a 1 for those elements $C_{Tot}[i, j] \geq k$ (idem for $D_{Tot}[i, j]$ and $CD_{Tot}[i, j]$) and 0 otherwise. In this way, we obtain an array of 90 binarized matrices (obviously NOT the same matrices as before) for which we compute the HIM distance from the initial adjacency matrix $Ad$. We will show the results coming from this method in section 4.3.

4.2.3 Second method

For the second type of pipeline, we started creating three arrays $(C[i, j, k], D[i, j, k], CD[i, j, k])$ of dimensions $i, j = 1, \ldots, 10$ and $k = 1, \ldots, 30$ where the generic storey $C[., k]$ of the array $C[i, j, k]$ is the (not yet binarized) matrix coming from the proceedings of Chapter 3 for the $k^{th}$ time series $TSM_k$ (the same holds for $D[., k]$ and $CD[., k]$).

After doing this, we compute the median between all the element $C[i, j, k]$ with $i$ and $j$ fixed. We redo this procedure for every $(i, j)$ and we obtain a final matrix $C_{median}$ (a median matrix). All this is obviously done also for the arrays $D[i, j, k]$ and $CD[i, j, k]$, finding $D_{median}$ and $CD_{median}$.

Then, the same procedure of Chapter 3 concerning the choice of quantiles is applied onto these three matrices. Once found the three best quantiles, the three matrices are binarized (always with the same proceedings). This method is a sort of inversion of the first described before, and we will show its results (better than those of the first method) in the next section.

4.3 Results

In this section we present the graphical results coming from the two methods described in subsections 4.2.2 and 4.2.3. For both methods, the DTW seems to be the worst way to infer networks, because it does not contribute much to the inference.

The important thing is that for both method, the combined way of analysis is the best choice that one can do in order to best infer gene regulatory networks. We will show how these methods works on real biological data in the Chapter 5.

4.3.1 First method results

Here we show the results for the method based on fixed quantiles coming from a mean over all quantiles of the table 3.3. It is widely visible that the DTW method is not able to contribute positively to the inference.
Figure 4.1: Reverse engineered graphs with corresponding optimal distances for the first method. HIM distances in Table 4.1.
4.3.2 Second method results

Results coming from the method of the three dimensional arrays are shown here. In this particular case the correlation and the combined ways produce the same output.

Figure 4.2: Reverse engineered graphs with corresponding optimal distances for the second method. HIM distances in Table 4.1.

4.4 Discussion

Here we have presented the application of methods shown in Chapter 3; synthetic data reflecting biological genic profiles were used as benchmark. In order to do so, we have used an open-source tool for in silico benchmark generation. We have then applied our combined method in two different way, obtaining considerable results. The second approach came up to be the best one, so in the next chapter we will apply it to real biological data (it will be applied also the first approach), coming from microarray analysis.
4.4. DISCUSSION

Figure 4.3: HIM distances against quantiles for the second method. The red curve represents Dtw methods; the black curve represents Correlation methods; green curve represents the combined method.

<table>
<thead>
<tr>
<th>HIM</th>
<th>Cor</th>
<th>Dtw</th>
<th>Cor after Dtw-shift</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 method</td>
<td>0.2579</td>
<td>0.2757</td>
<td>0.2567</td>
</tr>
<tr>
<td>2 method</td>
<td>0.1486</td>
<td>0.2761</td>
<td>0.1486</td>
</tr>
</tbody>
</table>

Table 4.1: HIM distances for both described methods (first and second) of the graphs in Figures 4.1 and 4.2, GNW.
CHAPTER 4. RESULTS ON BIOLOGICALLY-LIKE SYNTHETIC DATA
Chapter 5

Results on biological data

The aim of this chapter is to assess the validity of the two methods described in Chapter 4 on real biological data. In order to do this, as first way, we have chosen a well-known biological network: a subnet of the gene regulatory network of a Gram-negative bacterium called Escherichia coli (commonly abbreviated in E. coli). As second choice, we opted for a data set coming from an experiment investigating the importance of the diurnal cycle on the starch metabolism of Arabidopsis thaliana, a small annual plant [54].

5.1 SOS DNA repair system of Escherichia coli

The biological data we have chosen to validate our methods in Chapter 4 come from a subnet of the regulatory network of the E.coli, as said before. The chosen subnetwork is known as SOS DNA repair system of Escherichia coli and it is composed by 8 genes: uvdD, lexA, umuD, recA, uvrA, uvrY, ruvA, polB. For each of them, we are provided with the corresponding time series (50 time steps, interval of 6 minutes). These 8 time series form a single experiment. Our data set is composed by 4 of these experiments [55].

As initial adjacency matrix Ad, representing the SOS system, we have chosen the matrix shown in Figure 7 of article [56]. It is an inferred matrix for the system derived using the authors’ proposed method. We have then binarized it and made it symmetric (since our approach is designed for undirected graphs). Thus, we have set to 1 all elements different from zero (+, - and X), except for the two elements on the diagonal which we set to zero to remove self-loops (see figure 5.1). The results for both our methods (first and second sections of Chapter 4) applied on these biological data are shown in figures 5.1, 5.1 and 5.2.

---

1The data is available at the www.weizman.ac.il/mcb/UriAlon/ webpage (Prof. Uri Alon, Weizmann Institute of Science, Israel). Follow the link Download, and then the choose Data on SOS system.
5.1.1 Results

We report in this section the results obtained from our experiments. There are two important observations that we want to highlight: first, our combined method outperforms the single methods also on real biological data. The second important remark is that, even if the matrix Ad is just an inferred matrix from another approach, our resulting networks do not deviate much from the networks found in [56]. This supports the validity of our new combined method.

![Graphs for E. coli data set](image)

Figure 5.1: Reverse engineered graphs with corresponding optimal HIM distances for the first method. Application to the *E. coli* data set. HIM distances in Table 5.2.

5.2 *Arabidopsis thaliana* system

The biological data in this section come from experiments on the impact that the diurnal cycle has on the starch metabolism of the *Arabidopsis thaliana*. These data are expression profiles of 800 genes recorded in 11 different time
Table 5.1: (a) Ad initial matrix for the *E. coli*. (b) HIM vs quantiles for the second approach (described in Chapter 4) applied to the real data. The red curve represents the Dtw method; the black curve represents the Correlation method; green curve represents the combined method. Although the green and the black curve are overlapping, there is a quantile where the combined method is better than the correlation one.

<table>
<thead>
<tr>
<th></th>
<th>HIM</th>
<th>Cor</th>
<th>Dtw</th>
<th>Cor after Dtw-shift</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 method</td>
<td>0.0653</td>
<td>0.1903</td>
<td>0.0505</td>
<td></td>
</tr>
<tr>
<td>2 method</td>
<td>0.0505</td>
<td>0.1901</td>
<td>0.0505</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.2: HIM distances for both described methods (first and second) of the graphs in Figures 5.1 and 5.2, *E.coli*. 

5.2. ARABIDOPSIS THALIANA SYSTEM

(0 0 0 0 0 0 0 0)

(a)

(b)
Figure 5.2: Reverse engineered graphs with corresponding optimal HIM distances for the second method. Application to the *E. coli* data set. HIM distances in Table 5.2.
5.2. ARABIDOPSIS THALIANA SYSTEM

Table 5.3: (a) Ad initial matrix for the Arabidopsis thaliana. (b) HIM vs quantiles for the second approach (described in Chapter 4) applied to the real data of *Arabidopsis thaliana*. The red curve represents the Dtw method; the black curve represents the Correlation method; green curve represents the combined method.

\[
\begin{pmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
\end{pmatrix}
\]

(a) (b)

<table>
<thead>
<tr>
<th>HIM</th>
<th>Cor</th>
<th>Dtw</th>
<th>Cor after Dtw-shift</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 method</td>
<td>0.1372</td>
<td>0.1481</td>
<td>0.1080</td>
</tr>
<tr>
<td>2 method</td>
<td>0</td>
<td>0.1481</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.4: HIM distances for both described methods (first and second) of the graphs in Figures 5.3 and 5.4, *Arabidopsis thaliana*.

points [54]. They are constituted by two time series for each gene and they are available in the R package *GeneNet*, in a file named *arth800expr* [57]. We have decided to use just 10 genes among the all 800, in particular gene5, gene60, gene424, gene683, gene470, gene343, gene270, gene408, gene536 and gene390 (see Figure 1 in [54]). We have extracted the initial adjacency matrix Ad (see table 5.3) directly from the graph plotted in Figure 1 of [54].

5.2.1 Results

We show here the results obtained from our experiments. Both methods described in 4 support the validity of our new combined method. It is important for us to highlight that the HIM = 0 in the second method results for the correlation was expected: the graph from which we have extracted the Ad (Figure 1 in [54]) was created precisely with the correlation method. What is important in our opinion is that also our combined method infer a network with HIM = 0. This because the method recognize when it is appropriate to use the dtw and when not.
Figure 5.3: Reverse engineered graphs with corresponding optimal HIM distances for the first method. Application to the *Arabidopsis thaliana* data set. HIM distances in Table 5.4.
Figure 5.4: Reverse engineered graphs with corresponding optimal HIM distances for the second method. Application to the *Arabidopsis thaliana* data set. HIM distances in Table 5.4.
5.3 Discussion

The aim of this chapter was the application of methods shown in Chapter 4; biological data were used as benchmark. These data were chosen from microarray analysis data of the bacterium *Escherichia coli* and the small plant *Arabidopsis thaliana*. The second approach came up to be the best one, and we have also proved that (not only on synthetic data) our combined method is able to recognize whether the DTW shift should be used or not.
Chapter 6

Summary and Conclusions

The aim of systems biology is to develop mathematical models of biological systems. During the last years, many approaches have been developed to unravel the gene regulation complexities. The development of functional genomic technologies leads us to the ability to generate quantitative data representing the molecular state of cells at a genome level. Such datasets can be in the form of time series representing the dynamics of gene expression profiles in response to given stimuli, such as an environmental perturbation or the effect of a growth factor. Microarray technologies have gone a long way in this direction, and although recently new technologies are taking over (i.e., next generation sequencing), they remain an important source of genomic information. In spite of the relatively large amount of data information, inferring regulatory networks from observational data is still not trivial and is a matter of intense research. In this work, we have focused on the inference of gene regulatory networks from experimental data by means of computational methods.

A number of reverse-engineering approaches have been proposed. Some of these are designed to infer networks from a compendium of perturbation experiments while others are able to use time course data to generate dynamical models of gene interaction. Bayesian networks have been among the first to be applied to biological problems. They work by inferring probabilistic relationships between variables, can use either time course or steady state data and allow integration of prior knowledge in the model. Correlation-based methods compute correlation coefficients between variables to infer the underlying network topology. State-space models, and ODE-based methods, on the other hand, use time-course data to develop dynamic models of gene regulatory networks. For an extensive overview of these methodologies the reader is referred to [58].

Our work introduces a new method for network inference, which uses Dynamic Time Warping analysis (DTW) to integrate the correlation-based inference method. Here we show that our methodology is effective for a
quite wide spectrum of synthetic data sets. Moreover, the application of this method to real data sets (see chapter 5) has shown potential in identifying key regulators of important biological processes. The novelty of our approach derives from such synergetic combination of existing methods: we can claim that our combined method is better than each of the two methods separately considered.

Future research avenues include further investigation on how to combine existing inferring methods into more robust ones. As we mentioned in Chapter 1, one of the most important breakthrough in modern correlation world is representing by the MIC analysis (Maximal Information Coefficient). The combination of MIC and DTW approaches was not the aim of this thesis, but we are aware of the fact that this would be an important way of research. Moreover, it will be important to apply such new methodologies to more realistic networks, considering also the directionality of gene regulatory interactions. This would imply the employ of new mathematical tools, such as new network distances for directed graphs, in order to uncover causality within regulatory networks.

The inference network theory is surely an important area of research. The development of new technologies, such as next generation sequencing, will allow researchers to collect more specific data on genomic interaction. These technologies will also generate a considerably larger amount of data. Integrating such information within new inferring methodologies will be essential for a deeper understanding of how biological systems work.
Bibliography


network comparison via ipsen-mikhailov distance. 2011.


[47] D. Marbach, R. J. Prill, T. Schaffter, C. Mattiussi, D. Floreano, and 


*Journal of Computational Biology*, 16(2):229–239, 2009. WingX.


