ΕΛΛΗΝΙΚΟ
ΑΝΟΙΚΤΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ

Σχολή Θετικών Επιστημών και Τεχνολογίας

Μεταπτυχιακό Πρόγραμμα Σπουδών

Βιοπληροφορική και Νευροπληροφορική

Διπλωματική Εργασία

Μοντελοποιήση Γονιδιακών Ρυθμιστικών Δικτύων με Χρήση Μεθόδων Μηχανικής Μάθησης

Αργυρώ Καρατζή

Επιβλέπων καθηγητής: Παναγιώτης Βλάμος

Αθήνα, Ιούλιος 2022

ACKNOWLEDGEMENTS

I would like to acknowledge and give my thanks to my supervisor Professor Panayiotis Vlamos for all the help and his comments during the preparation of the present thesis.

ABSTRACT

One of the most useful mathematical tool in biological application is the network analysis. Networks can describe interactions between building blocks of organism such proteins, genes and metabolism. We focus on the role of networks in genes and particularly we deal with gene regulatory network reconstruction methods. Gene regulatory network inference has gained an increasing interest in the last few years mainly due to the vast amount of genetic information generated by new-generation approaches. Therefore, performing the back engineering task of identifying gene interactions based on a huge amount of gene expression data requires modern algorithm developed in the field of machine learning. Here, we perform a detailed description of the problem of gene regulatory network reconstruction focusing on the most recent and efficient machine learning methods employed in inference. We also describe analytically how a simulation analysis could be performed in order the efficiency of specific machine learning algorithms in Gene regulatory network inference to be tested.

Contents

List of Algorithms

List of Figures

Chapter 1

Introduction

The collection and analysis of network data plays a key role in a wide range of scientific fields. Nowadays, there is an explosion of data obtained from systems that can be conceptualized as networks. Examples include, but are not limited to, applications in biology, computer science, sociology and economics (Newman, 2012; Kolaczyk and Csárdi, 2014).

Here, we focus on the role of networks in biology. More precisely, the field of network analysis play a prominent role in understanding the functionalities of any organism. Therefore, network models are key tools in identifying interactions between biological elements which are building tools of living organism. In particular, network models are typically employed to model interaction between proteins (Protein-Protein networks), genes (Gene networks) as well as metabolites (metobolic networks). Our focus here are the Gene networks and particularly the, so-called, Gene Regulatory Networks (GRNs). A GRN can be described as the mechanism of maintaining life process, controlling biochemical reaction and regulating compound level, which plays an important role in various organisms and systems.

More specifically, we deal with the important process of GRN reconstruction. This process can be understood as a reverse engineering process where starting from gene expression data we attempt to identify the interaction between the genes and, thus, to understand the complexity, functionality and pathways of the biological systems. This in turn can have a huge impact in improving disease treatments by developing novel drugs. However, the recent advancements of microarray technologies and next generation sequencing, a huge amount of expression data is available. Therefore, GRN construction requires methods that can deal with the "big-data". These methods are typically developed in the field of machine learning.

In the present thesis we first a brief, but necessary, introduction to the notions of graph theory which underpins any network analysis and modelling. Then, we describe the main biological networks focusing on a comprehensive description of the GRNs. Subsequently, we describe the most recent and efficient algorithms that re typically employed in GRN reconstruction.

Chapter 2

Networks

2.1 Basic definitions and notation

Here we give basic definitions and notation that will be used throughout the thesis. In particular, we provide definitions for basic network concepts.

A graph (or network) is defined as $G = (V, E)$ where V is the set of vertices and $E \subset V \times V$ is the set of edges. In a directed graph every edge $(i, j) \in E$ links vertex i to vertex j (ordered pair of vertices) whereas if if $(i, j) \in E$ impies that $(j, i) \in E$ then the graph is called undirected. Every graph $G = (V, E)$ (directed or undirected) can be represented by its adjacency matrix $\boldsymbol{A}$. Matrix $\boldsymbol{A}$ has size $N \times N$, where N is the number of vertices in the graph, the rows and columns represent the vertices of the graph and the entries indicate the existence of edges. We write

$$a_{ij} = \begin{cases} w_{ij}, & \text{if } (i, j) \in E, \quad \forall\ i, j \in 1, \dots, N \\ 0, & \text{otherwise.} \end{cases}$$

In the case of unweighted graphs w_{ij} is a binary variable indicating the existence of an edge between the ith and jth vertex and in the case of weighted network w_{ij} is the weight of the edge. If the graph is undirected, the adjacency matrix $\boldsymbol{A}$ is symmetric, i.e., it is equal with its transpose $\boldsymbol{A}^{\top}$, for directed graphs the adjacency matrix is non-symmetric. A graph is called bipartite if the node set V can be partitioned into two disjoint sets V_h and V_a, where $V = V_h \cup V_a$, such that every edge $e \in E_b$ connects a node of V_h to a node of V_a, i.e., $e = (i, j) \in E \Rightarrow i \in V_h$ and $j \in V_a$. In other words, there are no edges between nodes of the same partition.

General Network Characteristics

A key concept in a graph is the *nodes degree*. In an undirected graph, the degree of any node is the number of the edges that end up on this particular node. The nodes of a directed graph are associated with an *in-degree* and an *out-degree*. The in-degree of the ith node, $i = 1, \dots, N$, is the number of incoming edges and the out-degree of the ith node is the number of outgoing edges, whereas it is easy to check that in the case of undirected graphs, the in-degree is equal to the out-degree. The *degree matrix* is defined as the diagonal $N \times N$ matrix $\mathbf{D}$, with the degree of each node in the main diagonal. For directed graphs we can define the in- and out- degrees matrices similarly.

A *path* in a graph is defined as a collection of nodes with the property that every consecutive pair of nodes in the sequence is connected by an edge. Two nodes $i, j \in V$ are called *connected* if there is a path from node i to node j. The above definitions can be extended to directed networks, where in a *directed path*, a directed edge should exist from each node of the sequence to the next node.

The *distance* between two nodes (e.g., metabolites in a metabolic network) is defined as the length of the shortest path between them. As shortest path we define the minimal number of edges that need to be traversed to reach node j from node i.

An undirected graph G is *connected*, if for every pair of nodes $i, j \in V$ a path exists from node i to node j. A directed graph is *strongly connected* if for every pair of nodes $i, j \in V$, there is a directed path from i to j and a directed path from j to i, whereas G is *connected* if for every pair of nodes $i, j \in V$, it contains a directed path from i to j or from j to i and, finally, G is *weakly connected* if by replacing the directed edges with undirected a connected graph is produced.

A *connected component* in an undirected graph is a maximal subgraph where every pair of nodes is connected by a path. For directed graphs, the notions of *strongly connected component* and *weakly connected component* can be defined. In the former case, similar to the definition of strong connectivity that we described earlier, the edge directionality is taken into consideration, while a weakly connected component requires the existence of a path between every pair of nodes in the maximal subgraph without considering edge directionality.

Network Centralities

Some of the most common questions arising in any network analysis are about the importance of each node, the identification of nodes serving as a hub as well as nodes that play the role of bridges between clusters (communities) of nodes. Such questions can be addressed by studying different definitions of *network centralities*. The simplest centrality measure of each node is the *degree centrality* (Bonacich, 1987) which is the degree of the node. A centrality measure able to identify important nodes that communicate quickly with other nodes within a graph is the *closeness centrality* (Sabidussi, 1966). The closeness centrality is defined as the inverse of the sum of the distances of the node from the others. Nodes that are bridges between commmunities can be found by calculating the *betweenness centrality* (Freeman, 1977) which is defined for the ith node as the total number of shortest paths from node i to node j that pass through node i divided by the total number of paths from node i to node j. Finally, we note that there are other centralities focusing on special characteristics of the graph. For example, the *eccentricity centrality* (Hage and Harary, 1995) indicates how much easy or difficult it is to access a node from any other node in the graph. The *eigenvector centrality* identifies nodes which are connected to important nodes and the *subgraph centrality* encodes information about the existence of a node in all subgraphs of the network.

Network models

Here we describe briefly the most popular models that have been utilised in network analysis in order to understand the topology of an observed network; whether the observed characteristics of the graph are specific or are following general graph patterns. In particular, we give details for the *Erdos-Renyi*, the *Watts-Strogaz* and the *Barabasi-Albert* models. A detailed presentation of these models as well as of other more specific ones can be found in West et al. (2001).

The *Erdos-Renyi* model is the most common model in the network theory and is mainly used to describe the topological characteristics of random graphs. More precisely, the model assumes that V nodes are randomly connected with probability $p = \frac{2E}{V(V-1)}$ and that the degrees distribution of the nodes is binomial; the probability that a node has degree deg is approximately equal to $e^{-\mathrm{deg}_{avg}} \frac{\mathrm{deg}_{avg}^{deg}}{\mathrm{deg}!}$. A notable property of *Erdos-Renyi* networks is that for number of vertices that tends to infinity ($V \rightarrow \infty$) the degree distribution shares more and more properties with the Poisson distribution. Moreover, the *Erdos-Renyi* networks are homogeneous in the sense that their vertices of similar degree; small *Erdos-Renyi* networks are similar to disconnected networks and for p approximately equal to $1/V$ the network has a big subnetwork that is consisted of the majority of the connections appearing in the whole network.

The *Watts-Strogatz* model accounts for networks in which any vertex can be reached from any other vertex by following a small path (in a small number of steps). Typical examples of networks in biology that exhibit this characteristic are the *metabolic networks* which we briefly describe in Section 2.2. The *Watts-Strogatz* networks are consisted of small communities; have high clustering coefficient and short average path length.

The *Barabasi-Albert* model is employed to describe the so-called *scale-free* networks; in *scale-free* networks the degree distribution follows a power law whereas the number of neighbors of any given node is not standard. The main characteristic of this type of networks is that evolve overtime and new edges are appearing randomly. Typical examples of biological networks that are well-described by the *Barabasi-Albert* model are the Protein-Protein networks; see Section 2.2 for more details.

2.2 Networks in biology

This Section presents the three main types of biological networks; namely the Protein-Protein networks, the Gene networks and the Metabolic networks. A more detailed presentation of biological networks can be found for example in Junker and Schreiber (2011).

2.2.1 Protein-Protein networks

A Protein-Protein network, also known as Protein-Protein *interaction* (PPI) network, in an organism can be described as the frame of its signal circuit. The PPIs intervene between the cellular processes and environmental, genetic signals. In particular, the PPI of an organism describes all the interactions of its cell proteins which in turn control its molecular and cellular functions. Therefore, understanding protein reductions can en-light the disease and healthy states of any organism. A PPI encodes all the information about the protein-protein interactome of an organism, i.e., the whole set of its protein-protein interactions.

To populate PPIs there available quite a lot of methods that can detect protein-protein interactions. These detection methods can be classified in two classes: experimental methods and computational methods. The experimental methods are either *biophysical* methods and are based on information gained from techniques like X-ray crystallography, NMR spectrocsopy and others or direct or indirect *high-throughput* methods which are mainly based on gene co-expression methods. The main drawbacks of the experimental methods are their cost and the fact that they are time consuming. On the other hand, the computational methods are based on empirical or theoretical predictions to infer new protein-protein networks.

By studying the topologiacl properties of the PPI networks it has been discovered that they are, independently of the organism, scale-free. Therefore, some hub proteins have a central role in the network by participating in the vast majority of the interactions whereas any non-hub proteins are part of a small fraction of the interactions. More information about PPIs and their detection can be found, for example, in Jones and Thornton (1996).

2.2.2 Gene networks

The genes of an organism produce, through the transcription process, products such the mRNA and proteins which in turn play a key role in important processes such as cell differentiation, cell survival and metabolism. Thus, there are two main types of gene networks that are of great biological interest: i) the gene regulatory networks (GNRs) which encode information about DNA-protein interactions; we describe GNRs in great detail in the next Chapter and ii) the gene co-expression networks which can be described as follows.

Before we describe a gene co-expression network it is useful to mention what is meant by the term *gene expression*. Gene expression is a biological process in which a gene product (usually a protein or RNA) is created by encoding information from a particular gene. The gene expression profiles of multiple individuals (e.g. cancer patients) constitute datasets known as microarray datasets. The existing information, in a microaaray dataset, about the co-expression of two different genes is summarized by a gene co-expression network. More precisely, such a network is an undirected graph in which the nodes correspond to genes and an edge between two nodes exist if there is significant co-expression between the corresponding genes. The gene co-expression networks are useful in biological applications since by identifying significantly co-expressed genes we reveal information about the pathway of the protein complex in which these genes belong to and in turn we can discover the role of the genes in a disease or in a treatment; see for example Stuart et al. (2003) and references therein for more details and examples on the role of gene co-expression networks in genetics. Fig

The construction of a gene co-expression network is usually conducted in two stages. First, a co-expression measure between each pair of genes in the dataset is calculated. The most common measures are i) the Pearson's correlation coefficient of the expression of the genes, ii) the Euclidean distance between gene expressions, iii) the level of the mutual information and iv) the Spearman's rank coefficient. Thus, a similarity matrix between the genes is constructed. Then, by choosing a significance threshold we set similarities above this threshold to be equal to one otherwise we set the corresponding similarity equal to zero. The resulting matrix is the adjacency matrix of the gene co-expression network; see Figure 2.1 for an example of gene co-expression network.

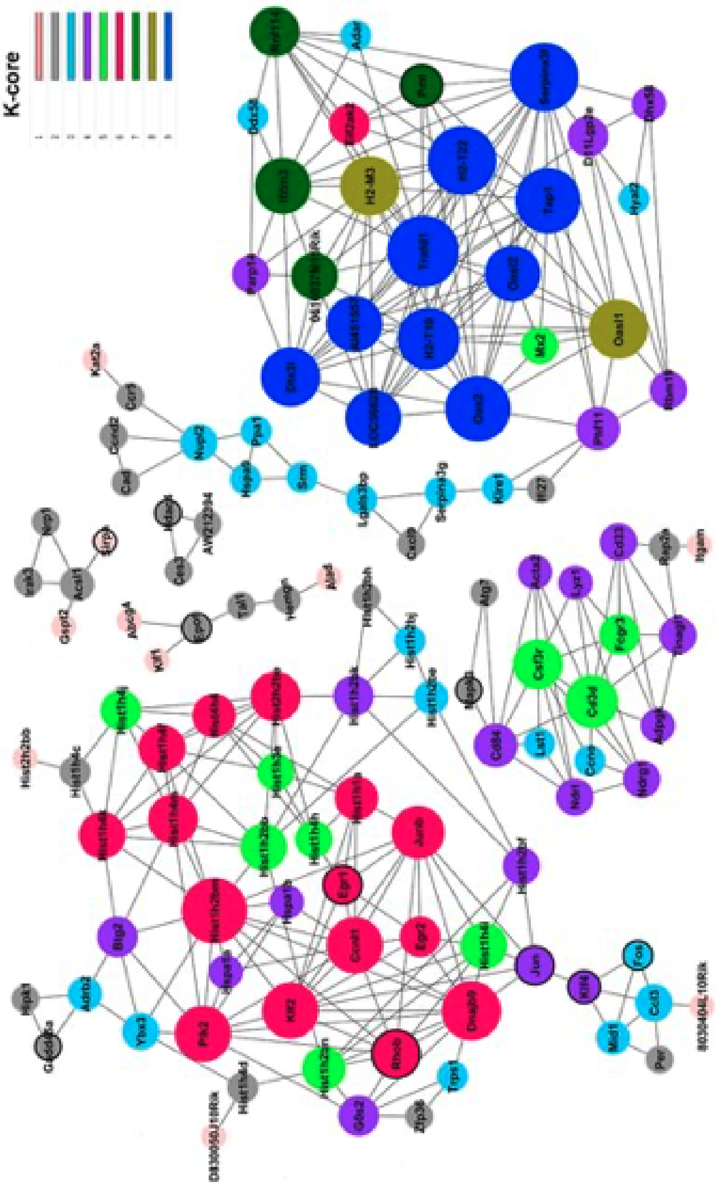

Figure 2.1: An example of a gene co-expression network; image taken from Zhang et al. (2011).

2.2.3 Metabolic networks

A metabolic network is consisted of interconnected pathways of biochemical reactions occuring in living cells. The nodes of a metabolic network are metabolites, i.e. end products of the metabolism in an organism, and its edges represent their known biochemical transformations. Therefore, a metabolic network encodes the all the physiological and biochemical properties of a cell. The metabolic networks are useful in the identification of cormobidity patterns in patients since disease phenotypes depend on the ability of a cell to breakdown. Thus, studying the complex interdependencies among a cell's molecular components can reveal deep functional and causal relationships between apparently distinct disease phenotypes cooccurring in the same organism; see for example in Lee et al. (2008) for the role of metabolic networks in the understanding of disease comorbidity. Figure 2.2 depicts an example of a human metabolic network.

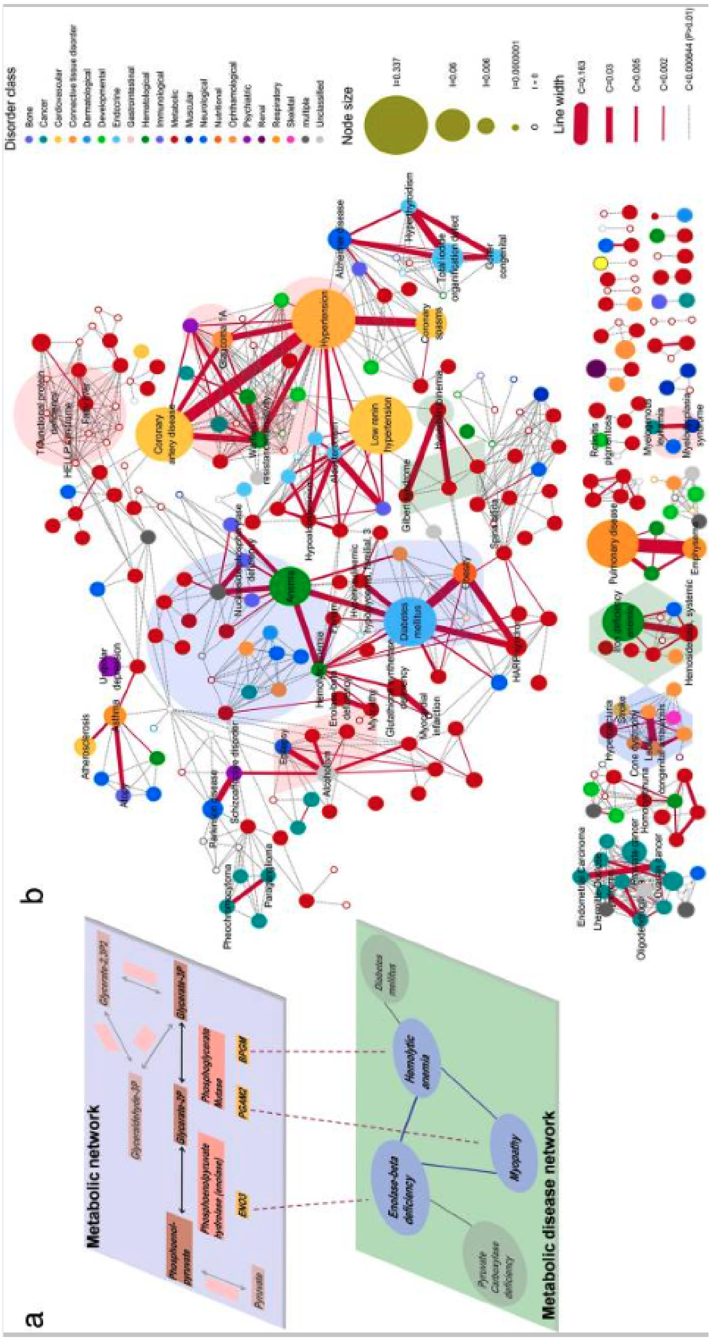

Figure 2.2: An example of a metabolic network in a human organism; image taken from Stuart et al. (2003).

Chapter 3

Gene Regulatory Networks (GRNs)

3.1 Overview

A GRN is consisted of the universe of molecular natures and encodes their interactions which in turn represent the function of a cell. In particular, GRNs describe cell-processes such the metabolism, the gene regulation as well as transport mechanisms. The nodes in GRN can represent genes, proteins, metabolites or RNA whereas the edges describe the molecular reactions between them. GRNs are are models which are typically used by researchers in order to discover new molecular interactions as well as diagnostic tools; see for example Liang et al. (2018).

Nowadays, GRNs have become on of the major tools, in the field of computational biology, utilised to understand and model complicated biological processes. The large development of GRN analysis has been mainly happened due to the amount of the gene expression information that is extracted in the very last few years. GRNs are proven to be one of the most efficient tools in the discovery of new connections between biological entities since quite a lot of the identified interactions have been confirmed experimentally; see for example Huang et al. (2018) for more details. The biological processes in which GRN inference has important contribution in their understanding range from from development to nutrition and metabolic coordination as well as in diverge fields including but not limited human health and agronomy (Levine and Davidson, 2005; Yan et al., 2016; Ogundijo et al., 2016).

An other aspect of GRN-based inference relies on the *reverse engineering* methods where GRN reconstruction is attempted out of experiemental results. The pipeline which is mainly used for such an analysis is commonly known as Knowledge Database Discovery (KDD) workflow. In the lines of KDD GRN reconstruction is going through inputa data pre-processing to the validation of the generating models. In the present thesis we follow the KDD workflow in order to show how GRN reconstruction is achieved as well as to perform related experiments. In the rest of this Chapter we first describe the biological data used for GRN inference as well as the mathematical models employed to perform reconstruction of GRNs. A quite large number of review papers provide details on both of the topics presented in the Chapter; see for example Delgado and Gómez-Vela (2019) and Zhao et al. (2021) which are two of the most recent review papers on the topic.

3.2 Mathematical modelling

3.2.1 Biological inputs for GRN inference

The development GRN reconstruction methods is highly connected with the development of technologies known as high-throughput. In particular, the rapid and on-going growth of the latter field allows quick advances in GRN inference as well. A quite recent sequencing tool known as next generation sequencing (developed by (Buermans and Den Dunnen, 2014)) has turned out to be the main source of gene expression data for GRN reconstruction; see for example Monger et al. (2015) and Pataskar and Tiwari (2016) for recent and detailed discussions. Importantly, by employing next generation sequencing methods we have available biological information from many different sources (e.g. multiple omics data) which in turn improve the efficiency of GRN inference. Figure 3.1 provides a visualisation of the GRN inference pipeline.

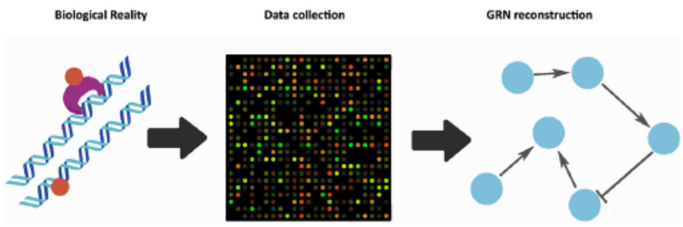

Figure 3.1: A representation of GRN inference: from biological data to network models; image taken from Delgado and Gómez-Vela (2019).

Omics in GRN reconstruction

The gene transcription process is broadly described (see e.g. Larvie et al. (2016)) as the key step in genes regulation. In particular, the majority of GRN methods attempt to identify direct or indirect relationships between transcript levels from omics datasets as well as to use existing biological knowledge in order to build robust models; describing the true biological connections as accurately as possible. Here, we describe two of the main omics datasets: the *Genome* and the *Transcriptome*.

The *Genome* of a biological system can be briefly described as its set of genes. These collections of genes range from protein-coding genes, which are the first that have been collected, to micro-RNAs and evolutionary-conserved regions. The most known nucleotide sequence databases are: the GenBank (USA), the EMBL (Europe) and the Data Bank of Japan Center (DDBJ); see in Benson et al. (2012) and Kodama et al. (2015) of their detailed description. An other source of Genome related data is the field of Epigenetics. For example, in Ramsey et al. (2018) they employ The Cancer Genome Atlas to construct a GRN for the identification of transcription factors with significant role in some cancers.

The term *Transcriptome* describes the analysis of gene expression patterns in order their relationships to be understood. It is well-known (see e.g. Lappalainen et al. (2013)) that gene expression levels are mainly governed by the transcription mechanism. In particular, non-coding RNA is one of the main genetic factors that drive the gene regulation process. See for example Parkinson et al. (2005), Clough and Barrett (2016), Kang et al. (2017) for more detailed discussions and related applications.

Data collection and pre-processing

The availability of the biological data required for GRN reconstruction relies on gene expression experiments. The latter are conducted frequently but the quality and the quantity of the generated data is not the same across the different experiments. Therefore, the first step is data collection with respect to GRN reconstruction is usually to build an experimental design. The design of the gene expression experiments introduces systematic perturbation on the observed biological system. Some of the most often perturbations include but are not limited to interventions at the transcriptomic, genetic, proteomic and metabolomic levels as well as changes in the environmental conditions. Then, by using a non-perturbated profile resulting from a presumed GRN we can evaluate the estimated goodness of the model. The effect of the experimental conditions can be measured either under equilibrium (steady-state) of the biological system by relying on *static* data or in a time-course situation where samples are drawn in a series of time points after perturbation.

The quality and the quantity of the data required for reliable GRN reconstruction depend on the information that we need to extract from the model. Reliable biological insights can generally provided by employing experimental data. However, in order to deal with drawbacks of relying on experimental data, e.g. bad quality and/or unavailability, those data are combined with external prior knowledge found in databases and in the related literature. Moreover, the unavailability of experimental data can also be treated by utilising fuzzy logic techniques to impute any missing data; see for example Bordon et al. (2015) for more details. Finally, it is important to note that the quality of the resulting model is not only determined by data quality but for the inference algorithm itself as well whereas there is also a clear correspondence between model complexity (dimensionality) and the amount of data required for an efficient GRN reconstruction.

After specifying an experimental design and understanding the data requirements with respect to the aim of the model data *pre-processing* is a necessary step for efficient GRN reconstruction. The aim of data *pre-processing* is to eliminate the two main sources of variability in GRN recostruction; systematic errors, referred also as bias in the data, and noise/stochastic effects. To remove systematic errors we usually perform data normalization while the data can be de-noised by considering several replicates to obtain repeated measurements of the variables of interest. Finally, we note that depending on the type (static or time-course) data further data pre-processing is may necessary; see for example Delgado and Gómez-Vela (2019) for more details.

3.2.2 Main models used for GRN reconstruction

GRN reconstruction relies on models which describe the nature of the regulatory dependencies among the biological organisms that belong to the networks underneath. In the present Section we describe the main modelling frameworks employed for GRN reconstruction; ordinal differential equations (ODE) models; Boolean networks; neural networks; Bayesian networks; information theory models.

ODE-based modelling

The ODE-based models provide the most accurate description of the gene expression dynamics by utilizing continuous variables. By denoting with g_j the expression level of the jth, among n in total genes, gene, $i = 1, \dots, n$, at time t, $t = 1, \dots, T$ the gene expression dynamics evolve over time according to the ODE

$$\frac{dg_j}{dt} = h_j(g_1, g_2, \dots, g_n, p, u),$$

u is a variable that accounts for external, environmental, factors, p denotes the parameters of the system and h is functional quantification of rate of changes of the states of the system. A complete specification of an ODE system also requires further specifications for the functions h_j as well as constraints that represent prior knowledge about the system. These specifications and constraints are necessary for the unique identification of the structure of the model and its parameters.

The main drawback of ODE-based models with respect GRN reconstruction is that they struggle to represent the highly complex non-linear dynamics of the regulatory processes. To that end, more flexible differential equation models have been developed which take into account the stochasticity of GRNs; these models are the stochastic differential equation models. More details about the application of ODE-based models in GRN reconstruction can be found, for example, in Matsumoto et al. (2017) and references therein.

Logical modelling

A popular tool for GRN inference is based on logical analysis and in in particular on Boolean networks. The Boolean networks are employed for GRN reconstruction since they can efficiently describe biological characteristics such as oscillation multi-stationary events, longrange correlations, switch-like behaviour stability and hysteresis. In a Boolean network each gene is represented by a variable and its expression level is indicated by a binary variable which classifies silenced or nearly silenced genes (low expression level) and activated genes (high expression level). Logical operators such as OR, AND and NOT are utilized to construct Boolean functions H which in turn reconstruct a directed graph $\mathcal{G}(G, H)$ where G is a variable associated with other variables as specified by the function H. Then, the state $g(t)$ of the network at time t writes

$$g(t) = g_{j,1}(t), g_{j,2}(t), \dots, g_{j,n}(t),$$

for all the nodes of the network, where j is referred to the jth gene.

The drawback of Boolean networks is the binary dicretization employed to classify activated and silenced genes. More precisely, a gene is rarely fully activated or deactivated and, in contrast, can have uncountable different states between these two extremes. Although the Boolean networks are considered as the simplest model for GRN reconstruction they are quite useful. See for example Simak et al. (2019), Claussen et al. (2017), Polak et al. (2017) and Moignard et al. (2015) for more details and related applications.

Deep learning algorithms

A quite recent direction in the GRN reconstruction field is the employment of deep learning methods for GRN inference. In particular, the deep learning algorithms commonly used for GRN reconstruction are the neural networks. A neural network is typically understood as a batch of algorithms that aims to identify the underlying relationships in a set of data by mimicking the way that the human brain operates. Therefore, neural networks refer to systems of neurons, either organic or artificial in nature. Two are the main Neural Network models which are usually employed in GRN reconstruction: the artificial neural network and the recurrent neural networks. The latter, in particular, allows the modelling of non-linear relationships and dynamic interactions across genes. The neural network models have the form

$$\frac{dg_j}{dt} = \frac{1}{\sigma_j}\left(h\big(\sum_{i=1}^{n}\beta_{ji}g_i + \nu_j\big) - \kappa_j u_j\right),$$

where β_{ji} encodes all the information about the relation of two genes in the jth and ith position, κ_j is decay rate parameter, ν_j denotes the basal expression level and u_j the gene expression level. The function accounts the regulatory effect on each gene and the weighted sum appearing within the function h is interpreted as as the regulatory effect on a particular gene. Finally, tasks such as evaluation of the outcomes, network performance optimization and error minimization are conducted by relying on some scoring function; see for example Kordmahalleh et al. (2017) for more details. Applications of neural network models in GRN inference can be found, among others, in Ling et al. (2013), Tong and Lin (2011) and Larkin et al. (2013).

Models based on directed acyclic graphs

One of the most popular modelling methods for GRN inference is based on the so-called directed acyclic graphs (DAGs) which are models developed under the Bayesian paradigm of statistical inference. The advantages of Bayesian methods in GRN reconstructions is that they combine probability theory (the Bayes theorem) with graph theory. Bayesian networks are typically understood as directed and acyclic graphs (DAGs), denoted by $\mathcal{G} = (G, A)$, and are accompanied with a set of probability distributions $\mathbb{P}$ that model the joint distribution of the nodes/genes $G = (g_1, \ldots, g_n)$; A refers to the directed rod corresponding to probabilistic dependency interactions between these genes.

The joint distribution of the nodes (variables) in DAG (Bayesian network) writes

$$\mathbb{P}(g_1, \ldots, g_n) = \prod_{i=1}^{n}\mathbb{P}(g_i|\text{parents}(g_i)),$$

where parents(g_i) denotes the set of all the *parent* nodes/genes regulating the child node g_i. A smart characteristic of Bayesian networks is the underlying Markov assumption: given its parents, each node is independent of its non-descendants. This assumption is convenient both in term of modelling as well as of computations.

Under the Bayesian networks paradigm and given some data D we try to identify the best DAG that describes the data by assigning to each graph $\mathcal{G}$

a score $s(\mathcal{G}, D)$ by employing the Bayes theorem (in the log-scale)

$$s(G, D) \propto \log \mathbb{P}(\mathcal{G}|D) + \log \mathbb{P}(\mathcal{G}),$$

where $\propto$ implies that the above relation is true with respect to a proportionality constant. The most common methods for learning a Bayesian network are consisted of three main steps Larjo et al. (2013): (i) Model selection, where DAGs are evaluated as candidate graphs of relationships; (ii) Parameter estimation: given graphs and experimental data sets identification of the best probability model for each node and (iii) Fitness rating: assign a score to each candidate model such that the higher the score, the better the model describes data. The latter is the model that represent the GRN learned from the data.

The main advantages of Bayesian methods and Bayesian networks, in particular, are (a) their ability to incorporate prior knowledge and (b) the fact that they can combine different types of data. Applications of Bayesian networks in GRN inference can be found, for example, in Acerbi et al. (2014), Chekouo et al. (2015) and Chudasama et al. (2018).

Information based network modelling

The most mathematically oriented models that are popular in GRN inference are based on the mathematical branch known as information theory. The information-theory based networks, also known as co-expression networks, exhibit high computational simplicity and thus are very popular tools in GRN inference. These types of networks identify relationships between pairs of genes by examining their dependence level; examining if this dependence level is above a pre-specified threshold. The dependencies between the genes are typically measured by using simple statistical measures of correlation; Pearson, Spearman or Kendall coefficients. More advanced measure based on Euclidean distances or mutual information, have been also applied for GRN reconstruction. Popular methods for GRN inference based on information theory have been developed by Liang et al. (1998), Butte and Kohane (1999), and Margolin et al. (2006) among others.

Chapter 4

Machine learning methods for GRN reconstruction

4.1 Machine learning algorithms

There is a vast variety of network reconstruction algorithms which use specific assumptions in order to deal with the uncertainty of processing. These assumptions can have an effect of their prediction accuracy. In the present Chapter we discuss modern machine learning methods that are commonly employed for GRN inference. The machine-learning methods typically consider the GRN reconstruction problem as classification or regression problem based on feature engineering and divide-and-conquer strategy. Then, a related machine learning algorithm is selected and finally the weights of the regulatory relationship identified are employed to rank and build the network. In what follows we describe three of the most common machine learning algorithms used for GRN reconstruction. We also note that the vast majority of the recently developed algorithm for GRN inference rely in one of the general methods described in the present Section: random forest, gradient boosting and support vector machines.

More precisely random forest methods have been applied, among others, on GRN reconstruction by Huynh-Thu et al. (2010) Petralia et al. (2015), Huynh-Thu and Geurts (2018) and Saremi and Amirmazlaghani (2021). Gradient boosting methods in GRN inference are currently one of the most popular techniques. There is a vast amount of the related literature where gradient boosting methods are employed either in the case of time-series or steady-state data. Examples include but are not limited to Slawek and Arodź (2013), Park et al. (2018), Moerman et al. (2019), Zheng et al. (2019) and Ma et al. (2020). Notice also that in the latter, recent, applications of gradient boosting in GRN inference the very efficient method of extreme gradient boosting (XGboost) is employed. Although details for the general gradient boosting can be found in the rest of the present Chapter, the interesting reader can find more details about XGboost in Chen and Guestrin (2016). Finally, support vector machines is also very popular machine learning technique in GRN inference. Early application of the support vector machine method in GRN reconstruction has been conducted by Ao and Palade (2011) and Yu et al. (2011) whereas Gillani et al. (2014) offer an early review of support vector machines in GRN inference. More recent applications of support vector machines in GRN reconstruction and inference include Ni et al. (2016), Razaghi-Moghadam and Nikoloski (2020), Chakraborty et al. (2021),Meher et al. (2021) and Yang et al. (2022).

4.2 Random forest

Random forest is a supervised learning algorithm which relies on an ensemble of decision trees, usually trained with the general technique of bootstrap aggregating or bagging. Here, we first describe briefly the building block of the random forest algorithm which are the decision trees. Then, we explain how is performed the preliminary step of decision tree learning. Finally, we give the details of the random forest algorithm which builds multiple decision trees and merges them together to get a more accurate and stable predictions.

Decision trees

The main building blocks of a random forest (classification or regression) algorithm are the decision trees. Figure 4.1 illustrates how two decision trees are built in order to assign scores in the individuals of a datasets based on a set of available features; age, sex and daily computer usage. From the visual inspection of the Figure it is clear how a decision tree works. We start by creating a node with respect to one of the available features and then we construct a path by separated in two directions by assigning a question tow the node. Then, in each one of the directions we either create an other node based on a different feature or we terminate the corresponding branch. More generally, a node in a tree can be thought as the point where the path splits into two — observations that meet the criteria go down the Yes branch and ones that do not go down the No branch. Finally, Figure 4.1 visualises also how the scores from the different trees are combined in order the desired score to be calculated. More precisely, random forests are a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance.

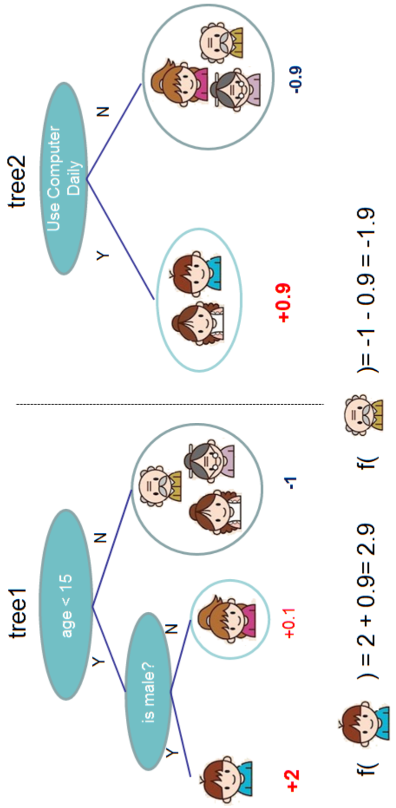

Figure 4.1: An example of two decision trees where two decision trees are combined in order a score to be assigned each individual of the dataset.

Bagging and random forest

To train a random forest algorithm the technique of bootstrap aggregating is applied to tree learners. In particular, let $X = x_1, \ldots, x_n$ be a training set with responses $Y = y_1, \ldots, y_n$. The main idea in bagging is to select B times a random sample with replacement of the training set and to fit trees to these samples; see Algorithm 1 for a summary of the steps.

Algorithm 1 Bagging.

```
Set the number of replications $B$.
for $b = 1, \ldots, B$ do
    Sample, with replacement, $n$ examples $X_b$ and $Y_b$ from $X$ and $Y$.
    Train a decision tree $f_b$ on $X_b$ and $Y_b$.
end for
```

To perform the 4th step of Algorithm 1 we can apply one of the most popular techniques of decision tree learning which is based on the concept of information gain through the notion of entropy[1]; see for example Wang and Suen (1984); Li et al. (2019) and references therein for more details.

After applying Algorithm 1 on the training data then based on a new data point x' a prediction y' can be calculated as

$$y' = \frac{1}{B} \sum_{b=1}^{B} f_b(x').$$

Notice also that an estimate of the uncertainty of the prediction can be made as the standard deviation of the predictions from all the individual regression trees on the new feature x'; see for example REF for more details.

Algorithm 1 summarizes the original bagging algorithm for trees. Random forests rely on an additional type of bagging. More precisely a modified tree learning algorithm is employed such taht at each candidate split in the learning process, a random subset of the features is selected. This process is usually called “feature bagging”. This procedure deals with the correlation of the trees in an ordinary bootstrap sample. In particular, if some features are accurate predictors for the response variable then, these features will be selected in many of the B trees which will be thus highly correlated.

4.3 Gradient boosting

Similarly to the decision forest algorithm presented in the previous Section the gradient boosting method is employed to conduct regression based GRN

[1] Entropy is the measure of uncertainty of a random variable, it characterizes the impurity of an arbitrary collection of examples. The higher the entropy more the information content.

inference. More precisely, there is a very recent interest in using the GRN literature to utilize gradient boosting techniques in order to perform GRN reconstruction; see for example Iglesias-Martinez et al. (2021) for on of the most recent approaches.

The gradient boosting method invented by Friedman (2001). The main idea in the gradient boosting method is to combine many "weak" learners in an accurate learners by performing several iterations. By imaging a liner regression setting we can describe the gradient boosting method as follows. In particular, we assume that we wish to learn a function/model F in order to obtain predictions $\hat{y} = F(x)$ by minimizing the mean squared error $(1/n)\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$. Then, a gradient boosting algorithm with B iterations works as follows. Let F_b, $b = 1, \ldots, B$ be a "weak" model in the sense that do no produce accurate enough iterations; for example for small b this model can be use the sample mean of the sample mean of the observations $\{y_i\}_{i=1}^{n}$ as a prediction $\hat{y}_i$ for each $i = 1, \ldots, n$. Then, in the next iteration of the gradient boost algorithm the model F_{b+1} can be an improved version of the model F_b by considering a new estimator $h_m(x)$. Therefore, we have that

$$F_{b+1}(x_i) = F_b(x_i) + h_b(x_i) = y_i.$$

The equation above implies that $h_b(x_i) = y_i - F_b(x_i)$ and, thus, the parameters of the model $h_b(x_i)$ can be learned by fitting h_b to the residuals $y_i - F_b(x_i)$.

More formally, the model h_b can be estimated by considering the loss function

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}(y_i - F_b(x_i))^2.$$

By noting the loss function above can be minimized by employing a gradient descent algorithm the gradient boosting method can be also considered as an optimisation algorithm where one has only to give as inputs the desired loss functiona and its gradients. Algorithm 2 summarises the steps of a general gradient boosting algorithm as presented by Wikipedia (Wikipedia contributors, 2022).

Algorithm 2 Gradient boosting.

```
**Input**: training set $\{x_i, y_i\}_{i=1}^{n}$; differentiable loss function $L(y, F(x))$, number of iterations $B$.
Set $F_0(x) = \arg\min_{\beta} \sum_{i=1}^{n} L(y_i, \beta)$
**for** $b = 1, \ldots, B$ **do**
    Compute the "pseudo-residuals"
        $r_{ib} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{b-1}(x)}, \quad i = 1, \ldots, n$
    Fit a simple model (e.g. decision tree) $h_b(x_i)$ to the training set $\{x_i, r_{ib}\}_{i=1}^{n}$.
    Set $\beta_b = \arg\min_{\beta} \sum_{i=1}^{n} L(y_i, F_{b-1}(x_i) + \beta h_b(x_i))$.
    Set $F_b(x) = F_{b-1}(x) + \beta_b h_b(x)$.
**end for**
Return $F_B(x)$.
```

4.4 Support vector machines

The last few years there is a lot of evidence that supervised machine learning methods outperform unsupervised and semi-supervised approaches for inference of GRN. This is because the identification of large number of transcription factors and their targets has enabled the availability of sufficient data to train supervised models; see for example Maetschke et al. (2014) for a detailed discussion. One of the most popular and efficient machine algorithm employed for supervised learning is the support vector machines (SVM) developed by Boser et al. (1992). Moreover, there is an increasing interest, in the literature related to GRN inference, in conducting GRN reconstruction by using SVM; see for example Ben-Hur and Noble (2005), Mordelet and Vert (2008), Khojasteh et al. (2021) and references therein for GRN reconstruction tools built the last 20 years. In the present Section we provide a brief presentation of the SVM techniques. We also note that there are several, public available, software applications which facilitate the employment of SVM; see Gillani et al. (2014) for software particularly designed to fit SVM in GRN reconstruction.

The main building block of SVM is a kernel function $k(x_i, x)$ which constructed to measure the similarity between the gene x and x_i. More precisely, for a dataset consisted of n genes a score to new gene x can be assigned by using the function

$$f(x) = \sum_{i=1}^{n} w_i k(x_i, x) + C,$$

where w_i are weights to be optimized in a training set and and C is the, so-called, complexity parameter which should be optimized to achieve an optimal predictive performance while it also controls ant overfitting of the training set. The aim during the training procedure is to classify the genes in the dataset in two classes; positive class with large positive scores and negative class with large negative scores. Finally, we note that there are a lot of choices for the kernel function k, Here, we summarize the most popular ones.

Linear and polynomial kernels

The linear kernel is the simplest kernel in SVM and is given by the formula $k(x, y) = x^\top y + c$. A polynomial kernel is a nonlinear kernel ideal for problem where all the training set is normalized. Therefore, the polynomial kernel is ccommonly employed in the case of microarray where the corresponding data are normalized by different normalization techniques before generating expression matrix. A polynomial kernel is defined by the formula $k(x, y) = (ax^\top y + c)^d$. Notice that the polynomial kernel has two additional parameters compared to the linear one, d denotes the degree of freedom (also known as order of polynomial) and a slope of alpha.

Gaussian and sigmoid kernels

The radial basis function is also known as a Gaussian kernel in the literature of SVM. This is a non-linear kernel given by

$$k(x, y) = \exp\{-\kappa ||x - y||^2\},$$

where $||\cdot||$ denotes a distance between the vectors which is usually chosen to be the Euclidean and κ is a parameters that controls the non-linearity of the kernel; if it is overestimated, it will behave almost as a linear kernel. The sigmoid (hyperbolic tangent) kernel is also known as multilayer perception kernel and is originated from the field of neural networks. It is given by the formula

$$k(x,y) = \tanh(ax^\top y + c),$$

where a and c are slope and intercept parameters respectively.

Chapter 5

Recent GRN methods

The aim of the present Chapter is to describe ongoing research on GRN reconstruction and inference. The main venues of current research in GRN reconstruction and inference are the following. A vast amount of recent research has been dedicated to evolutionary changes in GRN since it is nowadays well-recognized that regulatory changes play a major role in evolution of several species. The other modern direction of research in the field of GRNs that we discuss in the present Chapter is the update of the resources for GRN reconstruction and inference, i.e., the use of single-cell RNA sequencing in order to obtain gene expression data and overcome, thus, the shortcomings of conventional transcriptome sequencing technologies. Finally, at the end of the Chapter we discuss current challenges in the field of GRN inference.

5.1 GRN evolution

As early as in King and Wilson (1975) has been recognized that regulatory changes play a major role in evolution of species. More precisely, King and Wilson (1975) reached this conclusion by studying human-chimpanzee proteome similarities. This conclusion has been strengthen by further biological analysis; see for example Davidson (2010) for detailed discussion. In particular, it is now well-known that only a small number of genes shape the body plan of animals and these genes are parts of larger GRNs. Therefore, in order to understand evolution we need to study how GRNs evolve at the molecular scale. This requires well-constructed GRNs in more than one species. Moreover, the species under study have to be enough diverged enough such that there are genotypic and phenotypic differences but not so much diverged that regulatory modules cannot be identified.

Recently, Mehta et al. (2021) developed a novel computational pipeline in order to study the GRN evolution associated with phenotypic effect across ecologically diverse, vertebrate, species. Moreover, even more recently, Feigin et al. (2022) discuss how experiments and projects can be designed in order the GRN reconstruction and inference to play a prominent role in evolutionary biology.

5.2 GRN inference based on single-cell data

One of the latest contributions of GRNs in the biological understanding of living organisms is their utility in the employment of single-cell RNA-seq data for their construction. More precisely, the very recent years there is a large amount of research dedicated to employ single-cell RNA-seq snapshot data for the inference of the underlying GRNs. This approach is based on the fact that single-cell RNA sequencing captures the gene expression levels for a huge amount of individual cells in one experiment. Thus, the experimental design can be facilitated whereas large numbers of independent measurements, and accessing the interaction between the cell cycle and environmental responses that is hidden by population-level analysis of gene expression is also much easier.

The described approach of GRN reconstruction based on single-cell RNA-seq data relies mainly on moder machine learning methods. Here we present a small overview of the main methods used to conduct GRN inference based on single-cell RNA-seq data; see for example Pratapa et al. (2020) for a recent review of state-of-the-art algorithms employed in the recent literature to inferring GRNs from single-cell transcriptional data. Here we briefly described the most popular machine learning algorithms developed in the last decade to conduct GRN inference based on single-cell RNA-seq data.

5.2.1 Description of recent GRN inference based on single cell RNA-seq data

One of the first and well-known machine learning algorithms is the so-called `GENIE3` method developed by Huynh-Thu et al. (2010). The `GENIE3` algorithm constructs the regulatory network for each gene independently by employing tree-based ensemble methods in order to predict the expression level of each target gene from the expressions of all the other genes. The algorithm relies on the random forests methods described in Chapter 4.1. The importance of an input gene in the predictor for a target expression pattern determines the weight of the corresponding interaction. All these weighted interactions summed over all the genes consist of the regulatory network.

The second method that we describe here has been developed and implemented as an R-package (R Core Team, 2021) by Kim (2015); this is the r-package `PPCOR`. This package calculates for each pair of genes their the partial and semi-partial correlation with respect to the rest genes; a p-value for each correlation is also offered in the output of the package. This technique implies an undirected GRN in which the sign of the correlation, bounded between 1 and 1, can be used in order to signify whether an interaction is negative or positive.

The next method for GRN recosntruction based on single-cell transcriptional data that we describe has been become known as `pidc` and is based on information theory. The `pidc` technique partitions the pairwise mutual information between each pair of genes into a redundant and a unique component. Then, calculates the ratio between the unique component and the mutual information. The sum of this ratio over all other genes is the proportional unique contribution between the given pair of gene. This method has been developed by Chan et al. (2017).

A very popular method for GRN inference relying on single-cell RNA-seq data has been developed by Matsumoto et al. (2017) under the label `SCODE`. This particular techniques utilized linear ordinal differential equations in order to represent the transformation of a regulatory network in observed gene expression levels. The method relies on a specific relational expression estimated by employing simple linear regression. By combining the linear regression technique with a dimension reduction approach, `SCODE` results in a considerable reduction of the complexity of the constructed machine learning algorithm.

The technique `LEAP` developed by Specht and Li (2017) is also GRN recosntrction algorithm. The method is initialized with pseudotime-ordered data and then computes the Pearson's correlation of normalized mapped-read counts over temporal windows of a fixed size with different lags. Subsequently a maximum score for a pair of genes is stored the maximum Pearson's correlation over all the values of lag that the method considers. Finally, a permutation test is employed to estimate false discovery rates.

More recently than the described above methods the SINCERETIES algorithm developed by Papili Gao et al. (2018). The SINCERETIES algorithm is based on time-stamped transcriptional data in order to account for temporal changes the expression f each gene through the distance of the marginal distributions between two consecutive time points by employing the Kolmogorov–Smirnov statistical function. Regulatory connections for the target genes is achieved by relying on he Granger causality. More precisely, the SINCERETIES algorithm utilises the changes in the gene expression in a given time stamp in order to predict how the expression distributions of target genes shift in the next period of time. GRN inference is formulated as a ridge regression problem and partial correlation anaslysis determins the signs of the edges.

One more method that takes single-cell gene expression data over time course as input is the so-called SCNS algorithm constructed by Woodhouse et al. (2018). The developed algorithm calculates logical rules that inform the progression and transformation of initial cell states to later cell states. The resulting Boolean model is very useful for the prediction of the effect of gene perturbations on specific lineages.

A Bayesian method for GRN reconstruction by using single-cell RNA-seq data has been contributed by Sanchez-Castillo et al. (2018). This algorithm is known in the related literature with the name GRNVBEM and employs a first-order autoregressive model in order to estimate the fold change of a gene at a specific time. More precisely, under this approach this is expressed as a linear combination of the expression of the regulators of the gene in the a Bayesian directed acyclic graph at the previous time point. GRNVBEM constructed the underlying GRN relying on variational inference techniques; see for example Titsias (2009) for more details on Bayesian variational inference methods.

Recently, Deshpande et al. (2022) observed that although the majority GRN algorithms kick off by canclulationg a pseudotime value for each cell, the distribution of cells along the underlying dynamical process may not be uniform. Therefore, Deshpande et al. (2022) developed SINGE which utilises kernel-based Granger causality regression to alleviate irregularities in pseudotime values. More precisely, the developed technique conduct multiple regressions, one for each set of input parameters, and aggregates the resulting predictions using a variant of the Borda method.

To summarize, we need to mention that the majority of the algorithms and techniques described above are useful for single-cell transcriptomic data with cells ordered by pseudotime in the input. The described algorithms ideally require datasets corresponding to linear trajectories. However, some of the techniques suggest that data with branched trajectories can be split into multiple linear ones before input whereas theie majority construct finally a directed graph.

Chapter 6

GRN inference based on simulated data

In the present Chapter we show how a simulation study that highlights the benefits of utilizing modern machine learning methods in GRN reconstruction can be performed by using recently developed software tools. More precisely, the assessment of a GRN reconstruction method requires two ingredients. These are the graph structure of a GRN and gene expression data generated from that particular graph.

The network/graph structure of GRN can be obtained from already studied, well-known, interactions; typical examples include but are not limited to *E. Coli* (Santos-Zavaleta et al., 2019) and *S. cerevisiae* (MacIsaac et al., 2006). These type of data can be extracted from databases such as the, manually created, *RegulonDB* (Gama-Castro et al., 2016). On the other, hand gene expression data can be found in databases such as the *Gene Expression Omnibus* database (Clough and Barrett, 2016).

However, both the network structure of a GRN as well as gene expression data can be simulated. Here, we focus on probabilistic-based simulators. Simulation of gene expression data very often relies on a Gaussian models; see for example Danaher et al. (2014); Ha et al. (2015); Zhang et al. (2017) and Xu et al. (2018). Additional to Gaussian models raw RNA-seq count data are also simulated by using discrete probability models; examples here include Angly et al. (2012); Frazee et al. (2015) and Benidt and Nettleton (2015). Finally, the simulation of graphs which exhibit characteristics of network structure of a GRN has also gained the reference of the related literature. More precisely, recently the R-package (R Core Team, 2021) `SeqNet` (Grimes and Datta, 2021) constructs simulated networks with topology similar to real transcription networks.

In this Chapter we first discuss the employment of the R-package `SeqNet` in order to simulated both a network that exhibits topological characteristics similar to those of graphs of GRNs as well as to generated gene expression data according to the simulated network. Then, we show how a gradient-boosting decision tree (see e.g. Ke et al. (2017)) can be applied in order to perform GRN inference ad test how efficiently we reconstruct the simulated transcription network based on the corresponding gene expression data.

6.1 Simulation of the network structure

The network structure of a GRN can be represented by an undirectred graph in which each node corresponds to gene and an edge between two nodes exists if there is an association between the corresponding genes. The methods employed by the `SeqNet` package are based on representing the global network as a collection of overlapping modules which in turn represent regulatory pathways, i.e., set of interacting genes that regulate the production of mRNA and proteins.

The algorithm used by the `SeqNet` package to generate the network structure of a GRN works as follows. A global network $\mathcal{G} = (G, \{M^{(i)}\}_{j=1}^{\nu})$, where $G = \{1, \dots, p\}$ denotes the set of p genes and $M^{(j)} = (P^{(j)}, A^{(j)})$ is the ith module containing a subset $P^{(i)} \subset G$ of genes and $A^{(i)}$ is the adjacency matrix of the local network structure, is generated by iterating three steps: (i) generate randomly a module size, (ii) sample a set genes to construct the module, and (iii) generate the local adjacency matrix for the module. The steps (i)-(iii) are repeated such all the p genes have been utilized. In the following pragraph we discuss steps (i)-(iii) in more detail.

6.1.1 Details on network simulation

The first step in the procedure described above for the simulation of the network structure requires the generation of a random module size. This random size can be generated by a negative binomial (NB) distribution as follows. Let n_{min} a pre-specified minimum size for the module then we draw an integer $n \sim NB(n_{mod} - n_{min}, \eta)$, where n_{mod} denotes an average size of the module and η is a variance parameter. Then, the module size is set to be $n + n_{min}$; both n_{mod} and η are user specified.

In the second step of the network simulation algorithm discussed in the previous Section a set of genes is sampled in order to consist the module basis. Before describing the sampling procedure that is followed for the selection genes for each module we need to note the following. The aim of the designed procedure is twofold: First, the generated modules might have a non-empty intersection, i.e., they may be overlapping which implies that the same genes may be drawn multiple times. Second, in order to ensure that the global network is a single giant component that mimics the real transcription networks (see e.g. Dobrin et al. (2004)) we need to sample set of genes such that every new module is connected to at least one existing module. More precisely, to perform the sampling of the genes we have to work as follows. To generate the first module a subset of genes $P^{(j)} \subset G$ is drawn, after simulating a random module size as described in the previous paragraph, with equality probability $1/p$ for each gene in the set G which consisted of the indices of the p genes. Then, each additional module we work as follows. First, the random module size n is generated, then a node that links the current module with the previous ones is drawn randomly with probability that depends on the connectivity of each gene. Finally, the remaining $n - 1$ genes are drawn randomly with probability given to genes that already belong to a module.

The final, third step, of the procedure of network simulation generate the local adjacency matrix for each module. The generation of local netwrok structure for each module by the `SeqNet` package is based on the extremely popular Watts-Strogatz algorithm Watts and Strogatz (1998) which in turn relies of the well-known Barabasi-Albert model (Barabási and Albert, 1999). More precisely, the Watts-Strogatz algorithm proceeds as follows. The algorithm is initialized by network of p nodes in a ring lattice, with each node having initial degree 2ω and a probability π for edges to be rewired to a new neighbor. The case $\pi = 0$ maintains the ring lattice structure, the case $\pi = 1$ results in a random graph whereas for $0 < \pi < 1$ a small-world, not scale-free, is constructed. In particular, the `SeqNet` package simulates networks with a diverge range of topological structures beyond the scale-free one which although a common assumption is the analysis of gene expression data; see for example the seminal paper by Zhang and Horvath (2005) as well as Stumpf and Ingram (2005) and Parsana et al. (2019) for detailed discussions.

6.2 The network model

Notice that the simulated graph that represents the network structure of a GRN as described in Section 6.1 is an unweighted graph. In order to assign weights to each edge of the simulated network the algorithm constructed in the SeqNet package proceeds as follows. The process of adding weights is performed at the module level because the network is composed of individual modules and each one of them haa a local network structure. As already noted in the previous Section the existence of edges in each module represents the associations between the corresponding genes. In the SeqNet package the genes associations are modeled in conditional dependence basis. Therefore, a nonzero gene-gene association implies that the expression of two genes are conditionally dependent given the other genes in the module. Moreover, the non-existence between two genes (nodes) implies a zero association between them in the module's graph which in turn means that these two genes are conditionally independent. The described model for the genes associations can be formulated as probabilistic model based on the multivariate Gaussian distribution. Then, the described association model is widely known as a Gaussian graphical model (see e.g. Yin and Li (2011)) and the expression of p genes of the same module have the joint probability density function

$$\mathbb{P}(Z=z) = 2\pi^{-p/2}\det(K)^{1/2}\exp\{-(z-m)^\intercal K(z-m)/2\},$$

where m is the mean p-dimensional vector and K is a $p{\times}p$ positive definite matrix called the precision matrix which is the inverse of the covariance matrix usually employed to parametrize the normal distribution. The off-diagonal elements in K determine the conditional dependencies among genes. By noting that conditional dependence defines association in the described network model, this means that each gene-gene association can be represented through nonzero entries in K. In the SeqNet package there are efficient routines to generate a precision matrix and thus the association structure of the graph.

6.3 Simulation of RNA-seq data

As already discussed in the beginning of the Chapter the SeqNet package is consisted of three main components: he network generator, the Gaussian graphical model (GGM), and a converter from GGM values to RNA-seq expression data. In the previous Sections we described the first two components and here we describe the third one. In particular, in the present Section we discuss how the r-package `SeqNet` converges data generated from a GGM, as disccused in the previous Sections, to gene expression data. The generated expression data should fulfill two aims: (i) their dependence structure have to reflect the global network structure and (ii) the marginal distribution of each gene's expressions needs to be close enough to the reference RNA-seq data. Therefore, the procedure followed by the `SeqNet` package is developed in two parts: a) initial data are generated and aggregated together from the local GGMs defined in each module and b) the Gaussian values are transformed into RNA-seq data by sampling from the empirical distribution of the reference dataset. In what follows we describe algorithmically how these two parts are implemented by the `SeqNet` r-package.

- **Inputs**: A weighted graph $\mathcal{G} = (G, \{M^{(j)}\}_{j=1}^{\nu})$; a dataset $Y \in \mathbb{R}^q \times \mathbb{R}^q$ used a reference and consisted of q genes from $\tilde{n}$ samples; the desired sample size n.

- **Output**: A matrix $X \in \mathbb{R}^{n \times p}$ with the simulated RNA-seq expression data as generated from the given network.

(i) Draw p columns from the dataset Y where p is the number of nodes in $\mathcal{G}$.

(ii) Set $i = 1$ and initialize a matrix X with n rows and p columns.

(iii) Draw Gaussian random variable $\tilde{X} \in \mathbb{R}^p$ from $\mathcal{G}$.

(iv) Transform the Gaussian random values to $X_i = F_i(\Phi(\tilde{X}_i))$, where F_i is the emprical cumulative distribution function (cdf) of Y_i and Φ denotes the standard Gaussian cdf.

(v) Store the generated variables $X_1, \ldots, X_p$ in the ith row of X.

(vi) Repeat steps 3-5 for $i = 1, \ldots, n$.

Notice also that the procedure above can be implemented straightforwardly in several statistical software such as Python (vanRossum, 1995) and MATLAB (Higham and Higham, 2016).

6.4 GRN inference

In the previous Sections of the present Chapter we showed how statistical software can be utilised in order gene expression data from a given GRN structure can be simulated. In particular, we discussed how the r-package `SeqNet` can be used to perform such a simulation exercise. By noting that the simulated gene expression data are useful in order to test different (machine learning) methods with respect to GRN inference we describe, in the present Section, methods in order to conduct the final step that is required in such an analysis. This is to build the a GRN based on the simulated gene expression data and to compare the inferred GRN structure with the one we used to simulate the gene expression levels.

More precisely, we discuss how the recently developed, by Chen et al. (2021), r-package called Gene Network Estimation Tool (GNET) can be employed in order to construct GRNs from gene expression data. In particular, the methods employed by GNET rely on a probabilistic graphical model in order to conduct the GRN reconstruction. The GNET package include routines for data pre-processing, model development as well as visualization modules. We also note that the package summarizes and unifies previous work conducted by Zhu et al. (2012) and Zhu et al. (2013). In what follows we describe the inputs and outputs of the GNET package while we also briefly present the methodolgy in which the package is based on.

The r-package GNET is a module-based network method for GRN reconstruction and previous versions of the package have already been employed to infer theregulatory interactions among genes involved in several biological processes such as inference about regulations of estrogenes (Gong et al., 2014); see also Zhu et al. (2012) for more applications of the package.

The GNET tool employs techniques such as gradient boosting-based module initialization as well as visualization of experiment conditions in order to increase its functionality for the end user. The package treats a gene regulatory module as a two component object consisted of a regulatory tree (i.e. a binary decision tree) built from the gene expression profiles and a set of target genes regulated by the tree. Each node of the regulatory, binary, tree represents a regulator which is a transcription factor. Then, different branches of the binary tree are loaded with different samples from the input data based on the expression levels of the regulators in the tree nodes; the target genes within the same leaf node are assumed to share similar regulatory patterns.

In the initial step of the GNET package all genes except the regulators are clustered into groups; the number of clusters that are used for this step is user-defined. Then, a gradient-boosting decision tree is employed in order the different genes to be assigned to different initial clusters. Subsequently, an iterative regulatory tree-inference and gene re-assignment are conducted in order to develop the regulatory tree and update the target genes for each regulatory module. Finally, by using the GNET tools scores are assigned to the output modules in order the user to be able to identify biologically meaningful modules. These scores rely either on the similarity of genes in the same regulatory path or on the coherence between leaf nodes and user-defined labels of samples. Figure 6.1 illustrates the output of the GNET package after its application on the Arabidopsis RNA-Seq data.

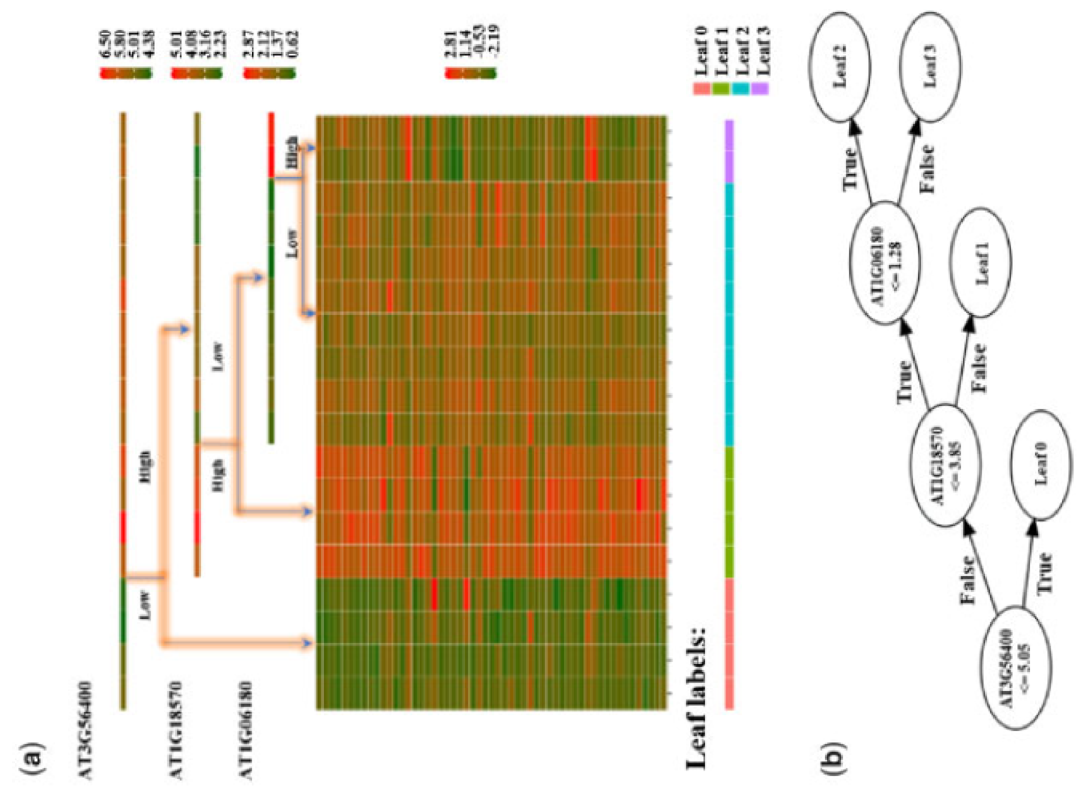

Figure 6.1: Visual output from the GNET package. (a) A regulator module generated from the Arabidopsis dataset. Top: Color bars indicating the separation of the samples according to the expression levels of the regulators. Middle: Heatmap which highlights the expression pattern for the target genes in the module. Bottom: Colar bar that visualizes the groups of samples that are in common leaf node of the tree. (b) The regulatory tree shown in (a); units of numbers in the log-scale. The figure is taken from Chen et al. (2021).

Chapter 7

Conclusions and Discussion

The focus of the present thesis was the role scientific fields such as network theory and machine learning estimation methods in the analysis of biological data. More precisely, we were interested in GRN reconstruction methods. Methods for the reconstruction of GRNs have received an increasing interest in the recent related literature mainly for two reasons:

(i) GRNs provide valuable infromation about pairwise interactions between biological entities. Therefore, they have been proven to be very successful in fields as human health and agronomy (Delgado and Gómez-Vela, 2019; Zhao et al., 2021) were, for example, can facilitate the understanding of the impact of complex diseases in cell functions as well as the development of biotechnological applications.

(ii) The recent years there is an explosion in the generation of genetic information through gene expression data. The large amount of the corresponding datasets has created the need for development of really efficient algorithms which are able to reconstruct GRNs utilising all the available information.

In the present thesis we first described the main and necessary tools from graph theory which are employed in the represantation and understanding of biological and/or molecular entities as networks. The, we briefly described the most common biological networks while we gave a detailed description of the GRNs. In particular, we presented data and pre-processing methods as well as the usual mathematical models employed in GRN reconstruction. Finally, we focused on the most recent machine learning methods for GRN reconstruction and we presented the pseudo-code that underpins their implementation.

Bibliography

Acerbi, E., T. Zelante, V. Narang, and F. Stella (2014). Gene network inference using continuous time bayesian networks: a comparative study and application to th17 cell differentiation. *BMC bioinformatics 15*(1), 1–27.

Angly, F. E., D. Willner, F. Rohwer, P. Hugenholtz, and G. W. Tyson (2012). Grinder: a versatile amplicon and shotgun sequence simulator. *Nucleic acids research 40*(12), e94–e94.

Ao, S. I. and V. Palade (2011). Ensemble of elman neural networks and support vector machines for reverse engineering of gene regulatory networks. *Applied Soft Computing 11*(2), 1718–1726.

Barabási, A.-L. and R. Albert (1999). Emergence of scaling in random networks. *science 286*(5439), 509–512.

Ben-Hur, A. and W. S. Noble (2005). Kernel methods for predicting protein–protein interactions. *Bioinformatics 21*(suppl_1), i38–i46.

Benidt, S. and D. Nettleton (2015). Simseq: a nonparametric approach to simulation of rna-sequence datasets. *Bioinformatics 31*(13), 2131–2140.

Benson, D. A., M. Cavanaugh, K. Clark, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and E. W. Sayers (2012). Genbank. *Nucleic acids research 41*(D1), D36–D42.

Bonacich, P. (1987). Power and centrality: A family of measures. *American journal of sociology 92*(5), 1170–1182.

Bordon, J., M. Movskon, N. Zimic, and M. Mraz (2015). Fuzzy logic as a computational tool for quantitative modelling of biological systems with uncertain kinetic data. *IEEE/ACM transactions on computational biology and bioinformatics 12*(5), 1199–1205.

Boser, B. E., I. M. Guyon, and V. N. Vapnik (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152.

Buermans, H. and J. Den Dunnen (2014). Next generation sequencing technology: advances and applications. *Biochimica et Biophysica Acta (BBA)-Molecular Basis of Disease 1842*(10), 1932–1941.

Butte, A. J. and I. S. Kohane (1999). Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. In *Biocomputing 2000*, pp. 418–429. World Scientific.

Chakraborty, A., S. Mitra, D. De, A. J. Pal, F. Ghaemi, A. Ahmadian, and M. Ferrara (2021). Determining protein–protein interaction using support vector machine: A review. *IEEE Access 9*, 12473–12490.

Chan, T. E., M. P. Stumpf, and A. C. Babtie (2017). Gene regulatory network inference from single-cell data using multivariate information measures. *Cell systems 5*(3), 251–267.

Chekouo, T., F. C. Stingo, J. D. Doecke, and K.-A. Do (2015). mirna–target gene regulatory networks: a bayesian integrative approach to biomarker selection with application to kidney cancer. *Biometrics 71*(2), 428–438.

Chen, C., J. Hou, X. Shi, H. Yang, J. A. Birchler, and J. Cheng (2021). Gnet2: an r package for constructing gene regulatory networks from transcriptomic data. *Bioinformatics 37*(14), 2068–2069.

Chen, T. and C. Guestrin (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794.

Chudasama, D., V. Bo, M. Hall, V. Anikin, J. Jeyaneethi, J. Gregory, G. Pados, A. Tucker, A. Harvey, R. Pink, et al. (2018). Identification of cancer biomarkers of prognostic value using specific gene regulatory networks (grn): a novel role of rad51ap1 for ovarian and lung cancers. *Carcinogenesis 39*(3), 407–417.

Claussen, J. C., J. Skiecevivcienė, J. Wang, P. Rausch, T. H. Karlsen, W. Lieb, J. F. Baines, A. Franke, and M.-T. Hütt (2017). Boolean analysis reveals systematic interactions among low-abundance species in the human gut microbiome. *PLoS computational biology 13*(6), e1005361.

Clough, E. and T. Barrett (2016). The gene expression omnibus database. In *Statistical genomics*, pp. 93–110. Springer.

Danaher, P., P. Wang, and D. M. Witten (2014). The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 76*(2), 373–397.

Davidson, E. H. (2010). *The regulatory genome: gene regulatory networks in development and evolution*. Elsevier.

Delgado, F. M. and F. Gómez-Vela (2019). Computational methods for gene regulatory networks reconstruction and analysis: A review. *Artificial intelligence in medicine 95*, 133–145.

Deshpande, A., L.-F. Chu, R. Stewart, and A. Gitter (2022). Network inference with granger causality ensembles on single-cell transcriptomics. *Cell reports 38*(6), 110333.

Dobrin, R., Q. K. Beg, A.-L. Barabási, and Z. N. Oltvai (2004). Aggregation of topological motifs in the escherichia coli transcriptional regulatory network. *BMC bioinformatics 5*(1), 1–8.

Feigin, C., S. Li, J. Moreno, and R. Mallarino (2022). The grn concept as a guide for evolutionary developmental biology. *Journal of Experimental Zoology Part B: Molecular and Developmental Evolution*.

Frazee, A. C., A. E. Jaffe, B. Langmead, and J. T. Leek (2015). Polyester: simulating rna-seq datasets with differential transcript expression. *Bioinformatics 31*(17), 2778–2784.

Freeman, L. C. (1977). A set of measures of centrality based on betweenness. *Sociometry*, 35–41.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.

Gama-Castro, S., H. Salgado, A. Santos-Zavaleta, D. Ledezma-Tejeida, L. Muñiz-Rascado, J. S. García-Sotelo, K. Alquicira-Hernández, I. Martínez-Flores, L. Pannier, J. A. Castro-Mondragón, et al. (2016). Regulondb version 9.0: high-level integration of gene regulation, coexpression, motif clustering and beyond. *Nucleic acids research 44*(D1), D133–D143.

Gillani, Z., M. S. H. Akash, M. Rahaman, and M. Chen (2014). Comparesvm: supervised, support vector machine (svm) inference of gene regularity networks. *BMC bioinformatics 15*(1), 1–7.

Gong, P., Z. Madak-Erdogan, J. Li, J. Cheng, C. M. Greenlief, W. Helferich, J. A. Katzenellenbogen, and B. S. Katzenellenbogen (2014). Transcriptomic analysis identifies gene networks regulated by estrogen receptor α (erα) and erβ that control distinct effects of different botanical estrogens. *Nuclear receptor signaling 12*(1), nrs–12001.

Grimes, T. and S. Datta (2021). Seqnet: An r package for generating gene-gene networks and simulating rna-seq data. *Journal of statistical software 98*(12).

Ha, M. J., V. Baladandayuthapani, and K.-A. Do (2015). Dingo: differential network analysis in genomics. *Bioinformatics 31*(21), 3413–3420.

Hage, P. and F. Harary (1995). Eccentricity and centrality in networks. *Social networks 17*(1), 57–63.

Higham, D. J. and N. J. Higham (2016). *MATLAB guide.* SIAM.

Huang, R., Y. He, B. Sun, and B. Liu (2018). Bioinformatic analysis identifies three potentially key differentially expressed genes in peripheral blood mononuclear cells of patients with takayasu's arteritis. *Cell Journal (Yakhteh) 19*(4), 647.

Huynh-Thu, V. A. and P. Geurts (2018). dyngenie3: dynamical genie3 for the inference of gene networks from time series expression data. *Scientific reports 8*(1), 1–12.

Huynh-Thu, V. A., A. Irrthum, L. Wehenkel, and P. Geurts (2010). Inferring regulatory networks from expression data using tree-based methods. *PloS one 5*(9), e12776.

Iglesias-Martinez, L. F., B. De Kegel, and W. Kolch (2021). Kboost: a new method to infer gene regulatory networks from gene expression data. *Scientific Reports 11*(1), 1–13.

Jones, S. and J. M. Thornton (1996). Principles of protein-protein interactions. *Proceedings of the National Academy of Sciences 93*(1), 13–20.

Junker, B. H. and F. Schreiber (2011). *Analysis of biological networks*. John Wiley & Sons.

Kang, T., W. Ding, L. Zhang, D. Ziemek, and K. Zarringhalam (2017). A biological network-based regularized artificial neural network model for robust phenotype prediction from gene expression data. *BMC bioinformatics 18*(1), 1–11.

Ke, G., Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems 30*.

Khojasteh, H., A. Khanteymoori, and M. H. Olyaee (2021). Engrnt: Inference of gene regulatory networks using ensemble methods and topological feature extraction. *Informatics in Medicine Unlocked 27*, 100773.

Kim, S. (2015). ppcor: an r package for a fast calculation to semi-partial correlation coefficients. *Communications for statistical applications and methods 22*(6), 665.

King, M.-C. and A. C. Wilson (1975). Evolution at two levels in humans and chimpanzees: Their macromolecules are so alike that regulatory mutations may account for their biological differences. *science 188*(4184), 107–116.

Kodama, Y., J. Mashima, T. Kosuge, T. Katayama, T. Fujisawa, E. Kaminuma, O. Ogasawara, K. Okubo, T. Takagi, and Y. Nakamura (2015). The ddbj japanese genotype-phenotype archive for genetic and phenotypic human data. *Nucleic Acids Research 43*(D1), D18–D22.

Kolaczyk, E. D. and G. Csárdi (2014). *Statistical analysis of network data with R*, Volume 65. Springer.

Kordmahalleh, M. M., M. G. Sefidmazgi, S. H. Harrison, and A. Homaifar (2017). Identifying time-delayed gene regulatory networks via an evolvable hierarchical recurrent neural network. *BioData mining 10*(1), 1–25.

Lappalainen, T., M. Sammeth, M. R. Friedländer, P. A. t Hoen, J. Monlong, M. A. Rivas, M. Gonzalez-Porta, N. Kurbatova, T. Griebel, P. G. Ferreira, et al. (2013). Transcriptome and genome sequencing uncovers functional variation in humans. *Nature 501*(7468), 506–511.

Larjo, A., I. Shmulevich, and H. Lähdesmäki (2013). Structure learning for bayesian networks as models of biological networks. In *Data Mining for Systems Biology*, pp. 35–45. Springer.

Larkin, A., L. K. Siddens, S. K. Krueger, S. C. Tilton, K. M. Waters, D. E. Williams, and W. M. Baird (2013). Application of a fuzzy neural network model in predicting polycyclic aromatic hydrocarbon-mediated perturbations of the cyp1b1 transcriptional regulatory network in mouse skin. *Toxicology and applied pharmacology 267*(2), 192–199.

Larvie, J. E., M. G. Sefidmazgi, A. Homaifar, S. H. Harrison, A. Karimoddini, and A. Guiseppi-Elie (2016). Stable gene regulatory network modeling from steady-state data. *Bioengineering 3*(2), 12.

Lee, D.-S., J. Park, K. Kay, N. A. Christakis, Z. N. Oltvai, and A.-L. Barabási (2008). The implications of human metabolic network topology for disease comorbidity. *Proceedings of the National Academy of Sciences 105*(29), 9880–9885.

Levine, M. and E. H. Davidson (2005). Gene regulatory networks for development. *Proceedings of the National Academy of Sciences 102*(14), 4936–4942.

Li, M., H. Xu, and Y. Deng (2019). Evidential decision tree based on belief entropy. *Entropy 21*(9), 897.

Liang, L., L. Gao, X.-P. Zou, M.-L. Huang, G. Chen, J.-J. Li, and X.-Y. Cai (2018). Diagnostic significance and potential function of mir-338-5p in hepatocellular carcinoma: A bioinformatics study with microarray and rna sequencing data. *Molecular medicine reports 17*(2), 2297–2312.

Liang, S., S. Fuhrman, and R. Somogyi (1998). Reveal, a general reverse engineering algorithm for inference of genetic network architectures. In *Biocomputing*, Volume 3.

Ling, H., S. Samarasinghe, and D. Kulasiri (2013). Novel recurrent neural network for modelling biological networks: oscillatory p53 interaction dynamics. *Biosystems 114*(3), 191–205.

Ma, B., M. Fang, and X. Jiao (2020). Inference of gene regulatory networks based on nonlinear ordinary differential equations. *Bioinformatics 36*(19), 4885–4893.

MacIsaac, K. D., T. Wang, D. B. Gordon, D. K. Gifford, G. D. Stormo, and E. Fraenkel (2006). An improved map of conserved regulatory sites for saccharomyces cerevisiae. *BMC bioinformatics 7*(1), 1–14.

Maetschke, S. R., P. B. Madhamshettiwar, M. J. Davis, and M. A. Ragan (2014). Supervised, semi-supervised and unsupervised inference of gene regulatory networks. *Briefings in bioinformatics 15*(2), 195–211.

Margolin, A. A., I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. Dalla Favera, and A. Califano (2006). Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. In *BMC bioinformatics*, Volume 7, pp. 1–15. BioMed Central.

Matsumoto, H., H. Kiryu, C. Furusawa, M. S. Ko, S. B. Ko, N. Gouda, T. Hayashi, and I. Nikaido (2017). Scode: an efficient regulatory network inference algorithm from single-cell rna-seq during differentiation. *Bioinformatics 33*(15), 2314–2321.

Meher, P. K., A. Mohapatra, S. Satpathy, A. Sharma, I. Saini, S. K. Pradhan, and A. Rai (2021). Predcrg: A computational method for recognition of plant circadian genes by employing support vector machine with laplace kernel. *Plant methods 17*(1), 1–15.

Mehta, T. K., C. Koch, W. Nash, S. A. Knaack, P. Sudhakar, M. Olbei, S. Bastkowski, L. Penso-Dolfin, T. Korcsmaros, W. Haerty, et al. (2021). Evolution of regulatory networks associated with traits under selection in cichlids. *Genome biology 22*(1), 1–28.

Moerman, T., S. Aibar Santos, C. Bravo González-Blas, J. Simm, Y. Moreau, J. Aerts, and S. Aerts (2019). Grnboost2 and arboreto: efficient and scalable inference of gene regulatory networks. *Bioinformatics 35*(12), 2159–2161.

Moignard, V., S. Woodhouse, L. Haghverdi, A. J. Lilly, Y. Tanaka, A. C. Wilkinson, F. Buettner, I. C. Macaulay, W. Jawaid, E. Diamanti, et al. (2015). Decoding the regulatory network of early blood development from single-cell gene expression measurements. *Nature biotechnology 33*(3), 269–276.

Monger, C., P. S. Kelly, C. Gallagher, M. Clynes, N. Barron, and C. Clarke (2015). Towards next generation cho cell biology: Bioinformatics methods for rna-seq-based expression profiling. *Biotechnology Journal 10*(7), 950–966.

Mordelet, F. and J.-P. Vert (2008). Sirene: supervised inference of regulatory networks. *Bioinformatics 24*(16), i76–i82.

Newman, M. E. (2012). Communities, modules and large-scale structure in networks. *Nature physics 8*(1), 25–31.

Ni, Y., D. Aghamirzaie, H. Elmarakeby, E. Collakova, S. Li, R. Grene, and L. S. Heath (2016). A machine learning approach to predict gene regulatory networks in seed development in arabidopsis. *Frontiers in plant science 7*, 1936.

Ogundijo, O. E., A. Elmas, and X. Wang (2016). Reverse engineering gene regulatory networks from measurement with missing values. *EURASIP Journal on Bioinformatics and Systems Biology 2017*(1), 1–11.

Papili Gao, N., S. M. Ud-Dean, O. Gandrillon, and R. Gunawan (2018). Sincerities: inferring gene regulatory networks from time-stamped single cell transcriptional expression profiles. *Bioinformatics 34*(2), 258–266.

Park, S., J. M. Kim, W. Shin, S. W. Han, M. Jeon, H. J. Jang, I.-S. Jang, and J. Kang (2018). Btnet: boosted tree based gene regulatory network inference algorithm using time-course measurement data. *BMC systems biology 12*(2), 69–77.

Parkinson, H., U. Sarkans, M. Shojatalab, N. Abeygunawardena, S. Contrino, R. Coulson, A. Farne, G. Garcia Lara, E. Holloway, M. Kapushesky, et al. (2005). Arrayexpress—a public repository for microarray gene expression data at the ebi. *Nucleic acids research 33*(suppl_1), D553–D555.

Parsana, P., C. Ruberman, A. E. Jaffe, M. C. Schatz, A. Battle, and J. T. Leek (2019). Addressing confounding artifacts in reconstruction of gene co-expression networks. *Genome biology 20*(1), 1–6.

Pataskar, A. and V. K. Tiwari (2016). Computational challenges in modeling gene regulatory events. *Transcription 7*(5), 188–195.

Petralia, F., P. Wang, J. Yang, and Z. Tu (2015). Integrative random forest for gene regulatory network inference. *Bioinformatics 31*(12), i197–i205.

Polak, M. E., C. Y. Ung, J. Masapust, T. C. Freeman, and M. R. Ardern-Jones (2017). Petri net computational modelling of langerhans cell interferon regulatory factor network predicts their role in t cell activation. *Scientific reports 7*(1), 1–13.

Pratapa, A., A. P. Jalihal, J. N. Law, A. Bharadwaj, and T. Murali (2020). Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nature methods 17*(2), 147–154.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Ramsey, J., K. Butnor, Z. Peng, T. Leclair, J. van der Velden, G. Stein, J. Lian, and C. M. Kinsey (2018). Loss of runx1 is associated with aggressive lung adenocarcinomas. *Journal of cellular physiology 233*(4), 3487–3497.

Razaghi-Moghadam, Z. and Z. Nikoloski (2020). Supervised learning of gene-regulatory networks based on graph distance profiles of transcriptomics data. *NPJ systems biology and applications 6*(1), 1–8.

Sabidussi, G. (1966). The centrality index of a graph. *Psychometrika 31*(4), 581–603.

Sanchez-Castillo, M., D. Blanco, I. M. Tienda-Luna, M. Carrion, and Y. Huang (2018). A bayesian framework for the inference of gene regulatory networks from time and pseudo-time series data. *Bioinformatics 34*(6), 964–970.

Santos-Zavaleta, A., H. Salgado, S. Gama-Castro, M. Sánchez-Pérez, L. Gómez-Romero, D. Ledezma-Tejeida, J. S. García-Sotelo, K. Alquicira-Hernández, L. J. Muñiz-Rascado, P. Peña-Loredo, et al. (2019). Regulondb v 10.5: tackling challenges to unify classic and high throughput knowledge of gene regulation in e. coli k-12. *Nucleic acids research 47*(D1), D212–D220.

Saremi, M. and M. Amirmazlaghani (2021). Reconstruction of gene regulatory networks using multiple datasets. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*.

Simak, M., H. H.-S. Lu, and J.-M. Yang (2019). Boolean function network analysis of time course liver transcriptome data to reveal novel circadian transcriptional regulators in mammals. *Journal of the Chinese Medical Association 82*(11), 872–880.

Slawek, J. and T. Arodź (2013). Ennet: inferring large gene regulatory networks from expression data using gradient boosting. *BMC systems biology 7*(1), 1–13.

Specht, A. T. and J. Li (2017). Leap: constructing gene co-expression networks for single-cell rna-sequencing data using pseudotime ordering. *Bioinformatics 33*(5), 764–766.

Stuart, J. M., E. Segal, D. Koller, and S. K. Kim (2003). A gene-coexpression network for global discovery of conserved genetic modules. *science 302*(5643), 249–255.

Stumpf, M. P. and P. J. Ingram (2005). Probability models for degree distributions of protein interaction networks. *EPL (Europhysics Letters) 71*(1), 152.

Titsias, M. (2009). Variational learning of inducing variables in sparse gaussian processes. In *Artificial intelligence and statistics*, pp. 567–574. PMLR.

Tong, H. and C.-Y. Lin (2011). Non-negative residual matrix factorization with application to graph anomaly detection. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, pp. 143–153. SIAM.

vanRossum, G. (1995). Python reference manual. *Department of Computer Science [CS]* (R 9525).

Wang, Q. R. and C. Y. Suen (1984). Analysis and design of a decision tree based on entropy reduction and its application to large character set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (4), 406–417.

Watts, D. J. and S. H. Strogatz (1998). Collective dynamics of 'small-world'networks. *nature 393*(6684), 440–442.

West, D. B. et al. (2001). *Introduction to graph theory*, Volume 2. Prentice hall Upper Saddle River.

Wikipedia contributors (2022). Gradient boosting — Wikipedia, the free encyclopedia. [Online; accessed 12-June-2022].

Woodhouse, S., N. Piterman, C. M. Wintersteiger, B. Göttgens, and J. Fisher (2018). Scns: a graphical tool for reconstructing executable regulatory networks from single-cell genomic data. *BMC systems biology 12*(1), 1–7.

Xu, T., L. Ou-Yang, X. Hu, and X.-F. Zhang (2018). Identifying gene network rewiring by integrating gene expression and gene network data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics 15*(6), 2079–2085.

Yan, W., W. Xue, J. Chen, and G. Hu (2016). Biological networks for cancer candidate biomarkers discovery. *Cancer Informatics 15*, CIN–S39458.

Yang, B., W. Bao, B. Chen, and D. Song (2022). Single_cell_grn: gene regulatory network identification based on supervised learning method and single-cell rna-seq data. *BioData Mining 15*(1), 1–18.

Yin, J. and H. Li (2011). A sparse conditional gaussian graphical model for analysis of genetical genomics data. *The annals of applied statistics 5*(4), 2630.

Yu, X., T. Liu, X. Zheng, Z. Yang, and J. Wang (2011). Prediction of regulatory interactions in arabidopsis using gene-expression data and support vector machines. *Plant Physiology and Biochemistry 49*(3), 280–283.

Zhang, B. and S. Horvath (2005). A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology 4*(1).

Zhang, J., Y. Yang, Y. Wang, J. Zhang, Z. Wang, M. Yin, and X. Shen (2011). Identification of hub genes related to the recovery phase of irradiation injury by microarray and integrated gene network analysis. *PloS one 6*(9), e24680.

Zhang, X.-F., L. Ou-Yang, and H. Yan (2017). Incorporating prior information into differential network analysis using non-paranormal graphical models. *Bioinformatics 33*(16), 2436–2445.

Zhao, M., W. He, J. Tang, Q. Zou, and F. Guo (2021). A comprehensive overview and critical evaluation of gene regulatory network inference technologies. *Briefings in Bioinformatics 22*(5), bbab009.

Zheng, R., M. Li, X. Chen, F.-X. Wu, Y. Pan, and J. Wang (2019). Bixgboost: a scalable, flexible boosting-based method for reconstructing gene regulatory networks. *Bioinformatics 35*(11), 1893–1900.

Zhu, M., J. L. Dahmen, G. Stacey, and J. Cheng (2013). Predicting gene regulatory networks of soybean nodulation from rna-seq transcriptome data. *BMC bioinformatics 14*(1), 1–13.

Zhu, M., X. Deng, T. Joshi, D. Xu, G. Stacey, and J. Cheng (2012). Reconstructing differentially co-expressed gene modules and regulatory networks of soybean cells. *BMC genomics 13*(1), 1–13.

Υπέθυνη Δήλωση Συγγραφεά:

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του N. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα εργασία αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον.