# University of Padova

Department of Mathematics "Tullio Levi-Civita"

*Master Thesis in Data Science*

# Machine Learning Approaches for Gene Regulatory Network Inference Using Single-Cell RNA Sequencing Data

*Supervisor*
Prof. Nicolò Navarin
University of Padova

*Co-supervisor*
Prof. Gabriele Sales
University of Padova

*Master Candidate*
Bogdana Živković

*Student ID*
2105083

*Academic Year*
2023-2024

# Abstract

Gene Regulatory Networks (GRNs) are essential for understanding the molecular interactions that drive biological processes, from development and metabolism to disease progression. GRNs can be represented as directed networks (or graphs), where nodes correspond to genes and directed edges indicate regulatory interactions between genes. GRN inference, the process of reconstructing these networks, has traditionally been performed using bulk RNA-sequencing (RNA-seq) data. However, the rise of single-cell RNA-sequencing (scRNA-seq) has introduced new opportunities and challenges, enabling the exploration of cellular heterogeneity at unprecedented resolution but also introducing significant technical noise and variability. Methods for GRN inference from scRNA-seq data can be classified into unsupervised and supervised approaches. Unsupervised methods identify regulatory interactions without prior knowledge of gene pairs, while supervised approaches rely on known networks to train models that predict gene interactions. This thesis investigates different machine learning approaches, GENIE3 and scGeneRAI, which are unsupervised, and GNNLink and STGRNS, which are supervised, for GRN inference using scRNA-seq data. Meaningful comparisons of methods were previously impossible because they were not originally tested on the same datasets. To address this, a significant contribution of this thesis is the evaluation and comparison of different methods using consistent datasets, ensuring direct comparability. While the performance of all methods is evaluated, specific enhancements were applied to GNNLink. These enhancements include using transcription factor frequency lookup tables to improve performance and creating an unsupervised version of GNNLink by leveraging only expression data to generate the training set based on Pearson correlation between genes. Additionally, irrelevant genes are filtered out from both the unsupervised approaches and the training set for the unsupervised version of GNNLink, ensuring that the predictions are not only more relevant but also comparable to those of supervised methods when evaluated against various ground-truth networks. By refining these computational methods, this research aims to improve the reliability and applicability of GRN inference across diverse biological contexts.

# Contents

# Listing of figures

# Listing of tables

# Listing of acronyms

**AUPRC** ..... Area Under the Precision-Recall Curve

**AUROC** ..... Area Under the Receiver Operating Characteristic Curve

**ChIP-seq** ..... Chromatin Immunoprecipitation Sequencing

**DNA** ..... Deoxyribonucleic Acid

**GCN** ..... Graph Convolutional Network

**GSD** ..... Gonadal Differentiation

**GNN** ..... Graph Neural Network

**GRN** ..... Gene Regulatory Network

**hESC** ..... Human Embryonic Stem Cell

**hHEP** ..... Human Hepatocyte-like Cell

**HSC** ..... Hematopoietic Stem Cell

**LRP** ..... Layerwise Relevance Propagation

**mDC** ..... Mouse Dendritic Cell

**mESC** ..... Mouse Embryonic Stem Cell

**MSE** ..... Mean Square Error

**mRNA** ..... Messenger Ribonucleic Acid

**RNA** ..... Ribonucleic Acid

**RNN** ..... Recurrent Neural Network

**scRNA-seq** ..... Single-cell RNA Sequencing

**TF** ..... Transcription Factor

**VSC** ..... Ventralized Spinal Cord

# 1 Introduction

This chapter outlines the key challenges encountered in gene regulatory network (GRN) inference, specifically the limitations of existing unsupervised and supervised methods. The chapter also presents the objectives of the thesis, which aim to address these challenges.

## 1.1 Problem Overview

Gene regulatory networks (GRNs) are used to illustrate the regulatory relationships between genes. Analyzing GRNs is crucial for understanding complex diseases, improving prevention, diagnosis, and treatment, and identifying new drug targets. The progress of single-cell RNA sequencing (scRNA-seq) technology has led to a drastic increase in single-cell gene expression data. This growth has created a pressing need for computational methods capable of utilizing these vast datasets to reveal potential gene interdependencies [13].

One of the ways to categorize methods for inferring gene regulatory networks from single-cell RNA sequencing data is by dividing them into unsupervised and supervised approaches. In the case of unsupervised approaches, statistical and computational techniques are used to uncover hidden patterns and structures within single-cell RNA sequencing data. This way, regulatory interactions can be identified without any prior knowledge of the gene pairs in the network. On the other hand, supervised approaches rely on a known network to train a model, which is then used to identify regulatory interactions between genes [14].

A key issue in analyzing different GRN methods is the absence of a comprehensive evaluation and comparison of these methods using the same datasets. Existing evaluations of these methods have often been conducted separately, using different types of data. Not all methods have been assessed using more biologically relevant experimental single-cell RNA sequencing data, which limits their applicability to real-world scenarios.

Another issue lies with the current evaluation of unsupervised methods, as they are often assessed without considering the distinction between relevant and irrelevant genes in the inferred gene pairs. Unsupervised methods infer gene pairs based solely on expression data, which may include genes not present in specific ground truth networks used for evaluation. To put it simply, some genes in an inferred pair may not be included in the ground truth, potentially affecting the evaluation. In transcription factor regulatory network (TRN) inference, it is essential to focus on filtering gene-gene relationships post-inference, particularly for unsupervised methods, so that only interactions where a transcription factor regulates a target gene are retained. This step ensures that predictions reflect biologically relevant regulatory dynamics. Similarly, since the ground-truth gene regulatory networks used for evaluation often include varying sets of genes, it becomes equally important to address this variability. Supervised learning approaches filter out irrelevant genes as the training datasets are curated for specific networks. However, unsupervised methods lack this filtering mechanism, which can lead to predictions that include irrelevant genes.

An issue encountered with supervised models for GRN inference is their traditional reliance on literature-derived datasets for training. These datasets consist of known gene pairs sourced from the same ground truth networks used for evaluation. Since biological knowledge and literature are constantly evolving, these datasets would require continuous updates. Moreover, they may be biased, as they often reflect only the currently known gene interactions, potentially overlooking undiscovered relationships. Furthermore, this approach may not be well-suited for real-world GRN inference cases, where known networks may not yet exist.

Finally, it has been observed that the previous supervised approaches have overlooked the prevalence of different transcription factors, and how this information could enhance GRN inference. Some transcription factors are more commonly involved in regulating other genes, a pattern that can be readily observed in ground truth networks. Failing to account for this variability can result in inaccurate predictions in GRN inference, as the influence of specific transcription factors may be either exaggerated or underestimated.

## 1.2 Objectives

Since the previous evaluations of methods were conducted using different datasets, making meaningful comparisons difficult, a key contribution of this thesis is the evaluation and comparison of various methods on consistent datasets, ensuring a direct and fair comparison. This thesis evaluates GENIE3 [11] and scGeneRAI [12] as representative unsupervised methods, and GNNLink [13] and STGRNS [14] as supervised approaches for gene regulatory network inference. Experimental scRNA-seq datasets along with their corresponding ground-truth networks are utilized to ensure that the evaluation of GRN inference algorithms aligns with realistic biological conditions. Lastly, the challenges outlined earlier are addressed through the following summarized approaches:

- **Refining Predictions by Addressing Variability in Ground-Truth GRNs**: To address the issue of irrelevant genes being present in inferred gene pairs, a post-inference filtering step was implemented in unsupervised methods that removes gene pairs that contain genes not relevant to the specific network being studied.
- **Overcoming Dataset Limitations with Expression Data**: An issue with supervised models for GRN inference is their reliance on constantly evolving literature-derived datasets. To overcome these limitations, it is advantageous to derive training datasets directly from expression data. By using gene-gene networks based on expression data, such as those inferred using correlation-based methods, the training and validation datasets are grounded in actual biological evidence. This solution could offer a more flexible and data-driven approach for model training.
- **Influence of TF Frequency on GRN Inference**: The proposed solution is to modify GNNLink by integrating a lookup table of transcription factor frequencies to adjust gene expression values. By incorporating TF frequency data into the inference process, predictions can potentially be enhanced.

This thesis is organized as follows: Chapter 2 provides the biological context for gene regulatory networks, gene expression, sequencing technologies, GRN inference methods, and performance metrics. Chapter 3 focuses on the datasets used in this study, highlighting the challenges of constructing ground-truth networks for GRN evaluation. Chapter 4 outlines the methods evaluated, including unsupervised approaches (GENIE3, scGeneRAI) and supervised models (GNNLink, STGRNS), along with their modifications and experimental setups. Chapter 5 presents the final evaluation results, for the original methods as well as their modified versions. The thesis concludes with a summary of key findings and their implications.

# 2
# Background

This chapter provides the necessary biological context for understanding the complexities of gene regulatory network (GRN) inference. It begins with an introduction to gene expression, laying the foundation for understanding how genes regulate various cellular processes. The concept of gene regulatory networks is then defined. Then, a key distinction is made between Bulk RNA sequencing and Single-cell RNA sequencing, highlighting their different applications in GRN analysis and their impact on the resolution of gene expression data. The chapter also discusses various methods used for GRN inference including correlation-based methods, regression-based approaches, probabilistic models, dynamical systems-based approaches as well as deep learning-based methods. Finally, the chapter addresses commonly used performance metrics in a biological context, with a focus on those suited for imbalanced data.

## 2.1 Biological Background

### 2.1.1 The Core Concepts of Gene Expression

The genome, often regarded as a fundamental element in the study of organisms, refers to the entirety of genetic information within a biological system [3]. Genetic information is represented by molecules called nucleic acids, which exist in two main forms: deoxyribonucleic acid (DNA) and ribonucleic acid (RNA). The genome of each cell is primarily stored in lengthy DNA molecules, each containing thousands of genes. Therefore, a gene can be described as a

linear segment of a DNA molecule. In contrast to DNA, RNA molecules are shorter and function to transmit genetic information to the cellular machinery, typically representing only one or a few genes. Both DNA and RNA are linear polymers composed of subunits known as nucleotides (Figure 2.1). The specific sequence of these nucleotides within each gene encodes the genetic information, with each type of nucleic acid containing four distinct nucleotides whose arrangement dictates this information. Each nucleotide consists of three main components: a phosphate group, a five-carbon sugar, and a nitrogenous base. In DNA, the sugar component is deoxyribose, while in RNA, it is ribose. Nucleotides can contain one of five different nitrogenous bases. DNA contains the bases adenine (A) and guanine (G), which are purines, and cytosine (C) and thymine (T), which are pyrimidines. In RNA, thymine (T) is replaced by uracil (U), so the bases are A, G, C, and U [1].

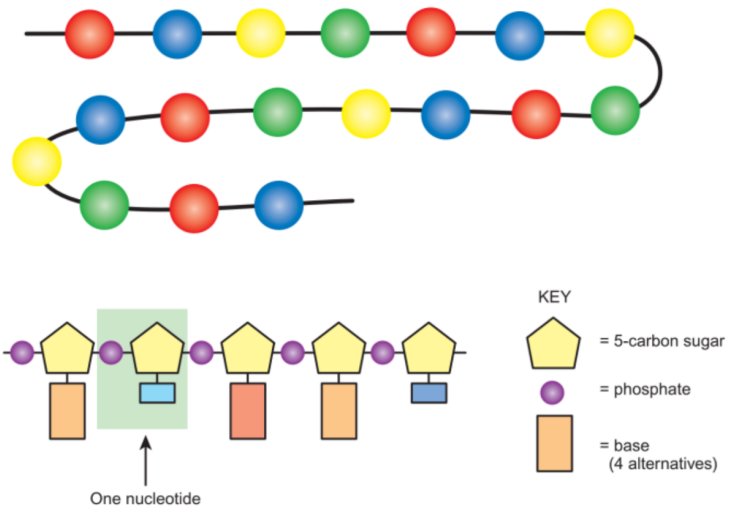


**Figure 2.1:** Genetic information is encoded by the specific sequence of nucleotides along a DNA or RNA strand. A nucleotide is made up of three main components: a phosphate group, a five-carbon sugar (either deoxyribose in DNA or ribose in RNA), and a nitrogenous base. Figure taken from [1]

According to the central dogma of molecular biology, the biology of living organisms can be interpreted in terms of the flow of information, which involves the transcription of genetic information encoded in DNA to messenger RNA (mRNA) by RNA polymerases, followed by the translation of mRNA to protein by ribosomes [15, 16]. This foundational concept provides the basis for understanding gene expression, which can be defined as the collective processes that result in specific levels of mRNA and protein within a cell. In various cell biology studies, gene expression serves as the basis for uncovering mechanisms at the microscopic

and molecular levels, while the gene expression profile functions as a detailed inventory at the macroscopic level [17]. Not all DNA within a cell is involved in coding for proteins or being transcribed. Protein coding genes make up a small fraction of the entire genome. One of the surprising findings from genome-sequencing projects around the turn of the millennium was the discovery that only about 3% of the human genome codes for proteins, with similar percentages seen in other higher organisms. Additionally, the number of genes remains relatively constant across different species, regardless of complexity. For example, the simple baker's yeast Saccharomyces cerevisiae has around 6,000 genes, which is more than a quarter of the number found in humans. This observation challenges human-centered perspectives and leads to the conclusion that the complexity of living organisms doesn't stem from the number of genes but rather from the interactions and dynamics between these genetic components [18].

While mRNA itself is not the final product of a gene, it serves as the initial step in gene regulation. mRNA transcript levels are essential for interpreting gene activity, as they indicate which genes are currently being transcribed and may be translated into proteins. Furthermore, measuring mRNA levels is currently more cost-effective than directly measuring protein levels and can be done in a high-throughput manner. Although the relationship between mRNA and protein abundance in cells can be complex and not always directly proportional, the absence of mRNA in a cell generally indicates low levels of the corresponding protein. Therefore, even though mRNA levels provide qualitative rather than quantitative estimates of the proteome, they still offer valuable insights into which proteins may be present or active under specific conditions [17].

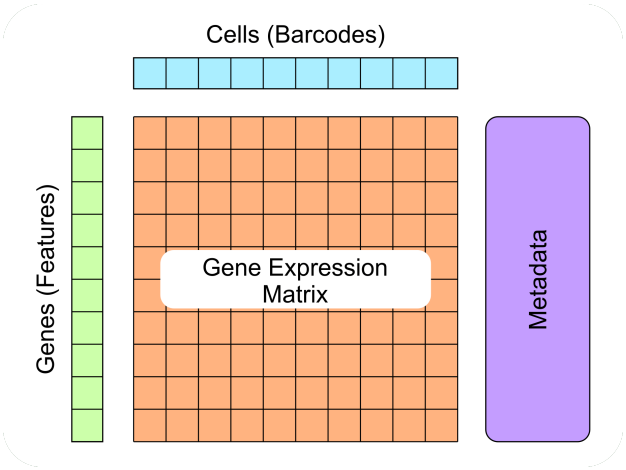


**Figure 2.2:** The gene expression matrix (colored orange) is a 2D matrix where rows represent genes (features), columns represent cells (barcodes), and the values indicate the gene expression levels for each cell. Figure taken from [2]

Gene expression profiles are created by assessing the transcription levels of genes across different conditions, developmental stages, and tissues within an organism. These profiles provide a comprehensive view of how each gene dynamically operates within the genome, revealing insights into their functional roles across varying biological contexts. Expression data (Figure 2.2) is typically structured in a matrix format, where genes are listed in rows, samples (such as various tissues, developmental stages, and treatments) are listed in columns, and each cell within the matrix represents the expression level of a specific gene in a specific sample [19].

Proteins constitute the essential structural and functional components within cells. There exists a variety of important roles, with each protein specialized to fulfill one of them, such as serving as a structural element, catalyzing enzymes, or functioning as antibodies. One of these critical roles is carried out by transcription factors (TFs), which are a specialized group of proteins. Transcription factors play a pivotal role in regulating gene expression and are associated with a wide array of diseases and phenotypes. These proteins bind to specific regulatory elements in DNA, such as promoters and enhancers, to either stimulate or inhibit gene transcription. By directly interacting with DNA, transcription factors control the formation of mRNA, thereby influencing the expression of genes and ultimately affecting cellular functions and processes. Due to regulatory proteins being products of expressed genes, they are also subject to regulatory mechanisms, leading to the development of complex networks of interacting genes [20, 21, 22].

### 2.1.2 Gene Regulatory Networks

Advances in biology have revealed that gene expression is controlled by complex networks crucial for cellular processes and organism complexity. These networks dynamically regulate gene expression to maintain individual phenotypes and adapt to environmental changes. Understanding transcriptional regulation, where transcription factors bind DNA to initiate gene transcription, is key to grasping gene expression control. Although transcription is a major control point, it's part of a broader mechanism that cells use to regulate their molecular functions and shape their phenotype [16]. At the heart of these complex regulatory mechanisms lie gene regulatory networks (GRNs), which comprise a set of molecular regulators that engage in interactions with one another and with various cellular components to regulate the expression levels of mRNA and proteins. These networks can be characterized as bipartite structures, where the nodes include genes and their regulators (such as protein-coding genes that encode transcription factors) that play roles in gene expression control [23].

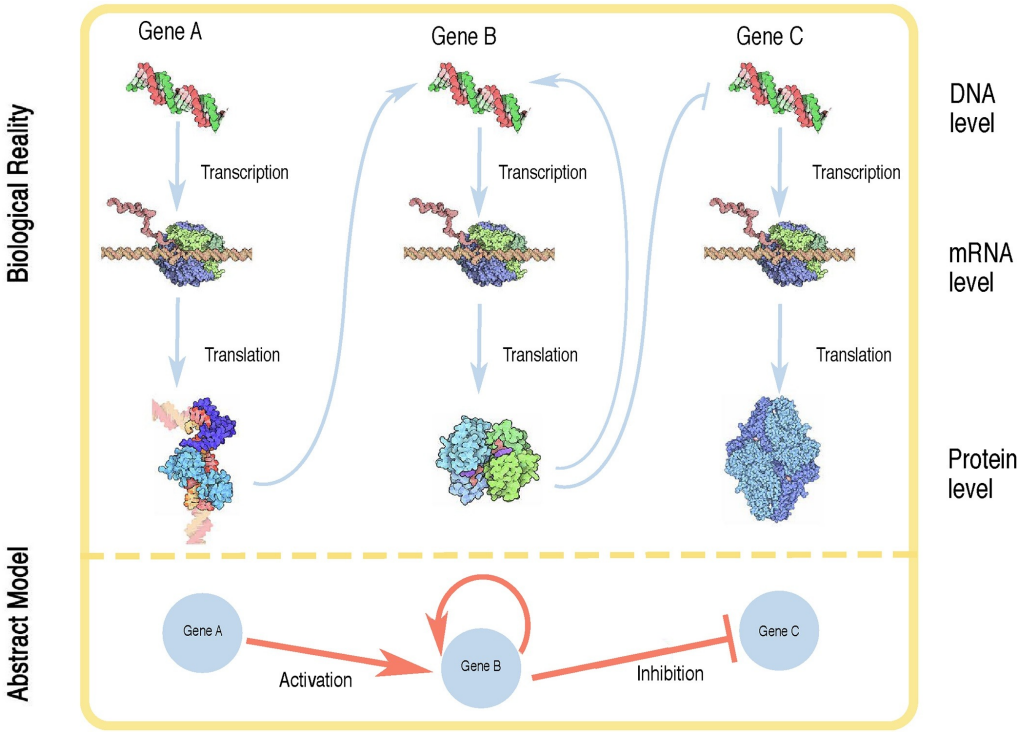


**Figure 2.3:** GRN inference aims to create abstract models that represent real biological processes. Figure taken from [3]

The relationships among the molecular regulators are often represented as directed edges, which elucidate the causal influences between regulatory nodes and their targets. Directed edges provide a clear causal relationship between a regulatory node (such as a transcription factor) and its target node, specifying which element influences the other. This directionality reflects the nature of regulatory interactions, where a specific regulator affects a particular target. Causality is typically determined through experimental methods like perturbation experiments, where altering one component (e.g., silencing a transcription factor) allows scientists to observe changes in its targets. However, when GRNs are derived from large-scale data sets, such as those based on gene expression correlations, the directionality of relationships is not always clear. Correlation between gene expression levels does not necessarily imply a causal link or indicate the direction of influence. To establish causality, additional quantitative approaches and methodologies are required [16]. Modeling these networks, as illustrated in Figure 2.3, presents a significant challenge. However, by accomplishing this goal, our understanding of cellular functions and insights into effective intervention strategies for treating human diseases are substantially enriched. This has driven numerous researchers to create network inference methods, also known as reverse engineering techniques, aimed at efficiently reconstructing

gene regulatory networks from expression data. The framework for GRN reconstruction is shown in Figure 2.4 [24].

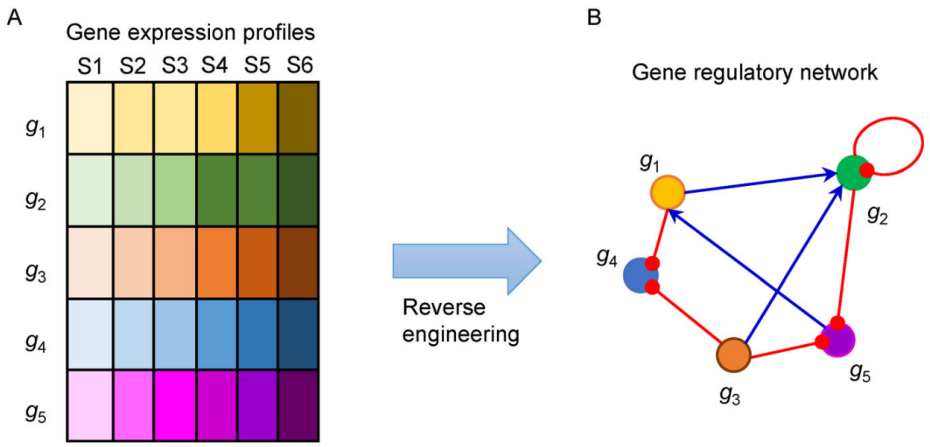


**Figure 2.4:** The framework of reconstructing gene regulatory network (B) from gene expression profiling data (A). Figure taken from [4]

By providing a comprehensive framework for understanding how different molecular entities influence each other, GRNs enable scientists to uncover new interactions, validate experimental hypotheses, and explore the underlying mechanisms driving biological processes. GRNs have proven to be revolutionary tools for identifying novel interactions between biological entities, which significantly aids in research and hypothesis formulation. They have demonstrated their effectiveness in various applications, including diagnostics, where many of their predictions have been experimentally validated, underscoring their reliability. GRNs are crucial for studying essential biological processes, from development and nutrition to metabolic coordination. Their implementation has led to advancements in human health and agriculture by facilitating the management and coordination of physiological events related to GRN activity. This includes applications in disease monitoring, biotechnology, and crop production. Additionally, GRNs have enhanced our understanding of developmental processes by illustrating how these networks generate developmental patterns [3].

### 2.1.3 Gene Regulatory Network Definitions

The definition of gene regulatory network (GRN) varies across studies (there is no clear consensus on the terminology used). Gene regulatory network is typically defined as a collection

of directed interactions between genes, where edges represent regulatory relationships. These edges point from a regulator gene to the gene being regulated, indicating the flow of regulatory influence. Since genes themselves are not biologically functional, gene regulation is carried out by the products of regulator genes, such as proteins or other molecules that are produced by these genes. These regulatory products influence the expression of other genes. By describing edges as directed, GRNs establish a clear distinction from gene co-expression networks. While gene co-expression networks involve genes as nodes with undirected edges representing co-expression relationships, GRNs feature directed edges that indicate regulatory interactions. This directional aspect allows GRNs to provide insight into causality, showing whether one gene regulates another, unlike gene co-expression networks, which do not reveal this regulatory dynamic. Although gene co-expression networks help identify genes with related functions, they lack the capacity to clarify whether the genes are co-regulated or influenced by another gene. In the literature, the term Gene Regulatory Network (GRN) is sometimes used broadly to refer to any network that models gene interactions, including those specifically involving transcriptional regulation. However, transcriptional regulatory networks (TRNs) are a more specific subset, where the edges exclusively represent the regulatory interactions involving transcription factors (TFs) and their target genes, focusing on transcriptional regulation [5]. In this thesis, the term 'Gene Regulatory Network' (GRN) will specifically denote Transcriptional Regulatory Networks (TRNs), focusing on the regulatory interactions where transcription factors (TFs) control target gene expression. Throughout the thesis, any reference to 'GRNs' should be understood as encompassing only this transcriptional regulatory context.

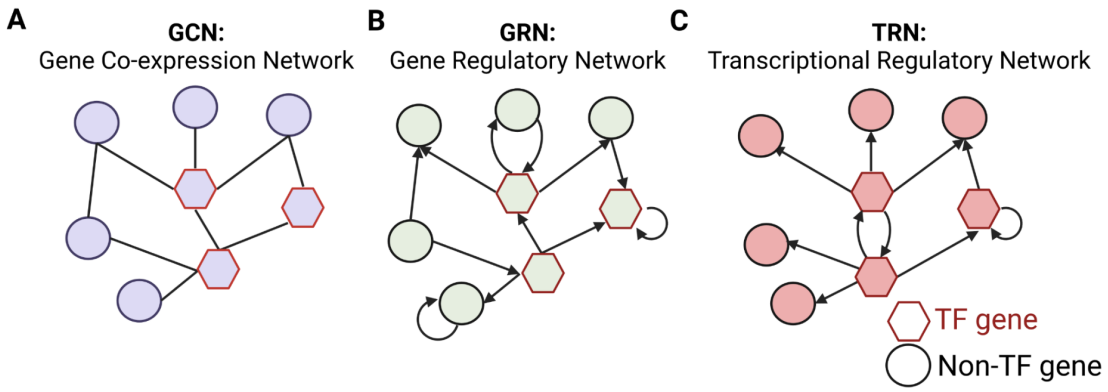


**Figure 2.5:** Types of gene networks include: A) Gene Co-expression Networks (GCNs), which contain undirected edges; B) Gene Regulatory Networks (GRNs), where edges are directed to indicate regulatory influence; and C) Transcriptional Regulatory Networks (TRNs), which are directed networks where edges can only originate from transcription factors (TFs). Figure taken from [5].

### 2.1.4 Single-cell RNA Sequencing

Bulk RNA sequencing (bulk RNA-seq) is one of the earliest and most commonly used RNA sequencing techniques in life sciences. It involves sequencing RNA extracted from a large population of cells, providing an averaged view of gene expression across the sample. Bulk RNA-seq has been widely applied in fields like cancer research, drug development, and diagnostics due to its ability to reveal gene activity in tissues or cell populations. While powerful, it lacks the ability to capture the nuances of individual cell behavior, a limitation that has driven the development of more refined RNA sequencing techniques [25]. Novel opportunities have emerged with the introduction of single-cell RNA sequencing (scRNA-seq), enabling the exploration of gene expression profiles at the single-cell level. Single-cell RNA sequencing has become increasingly favored for investigating fundamental biological questions related to cell heterogeneity and early embryo development, particularly in scenarios involving a small number of cells. This preference stems from the limitation of traditional bulk RNA sequencing, which predominantly reflects the average gene expression across thousands of cells and thus is less suited for these specific cases. In recent years, scRNA-seq has been widely used across various species, particularly in human tissues, both normal and cancerous, uncovering significant cell-to-cell gene expression variability. With advancements in sequencing technologies, several new scRNA-seq methods have been developed. These new protocols have significantly improved our ability to study how gene expression changes dynamically within individual cells, providing a much clearer and more detailed understanding of cellular processes. Each method has its own advantages and limitations, requiring careful selection based on research goals and sequencing costs [26]. The choice of scRNA-seq protocol depends on the specific research question. While most protocols accurately determine transcript abundance, they vary in sensitivity, affecting the detection of weakly expressed genes [27]. Low capture efficiency and high dropouts are challenges commonly associated with scRNA-seq caused by the limited amount of starting material. In contrast to bulk RNA sequencing, single-cell RNA sequencing (scRNA-seq) yields data with greater noise and variability. The presence of technical noise and biological fluctuations, such as stochastic transcription, introduces significant challenges for the computational analysis of scRNA-seq data. Many tools have been created for analyzing bulk RNA-seq data, but most of these methods aren't suitable for scRNA-seq data. While short-read mapping techniques can be used for both types of data, other analyses such as differential expression, cell clustering, and gene regulatory network inference differ between scRNA-seq and bulk RNA-seq. Because scRNA-seq data often contains high technical noise, quality control is essential to remove low-

quality data and ensure reliable results. As more tools are developed specifically for scRNA-seq, each with its own strengths and limitations, it is important to carefully choose the right analytical methods to manage the high variability in scRNA-seq data effectively [26]. Gene regulatory network inference is commonly performed in bulk RNA-seq studies using tools like weighted gene co-expression network analysis (WGCNA) [28] to construct networks based on gene co-expression. With single-cell RNA sequencing (scRNA-seq), similar approaches can be applied by treating individual cells as samples, potentially revealing new insights into gene correlations and regulatory relationships. However, due to technical noise and cellular heterogeneity, methods for network reconstruction in scRNA-seq must account for these factors to ensure robust and accurate results [26].

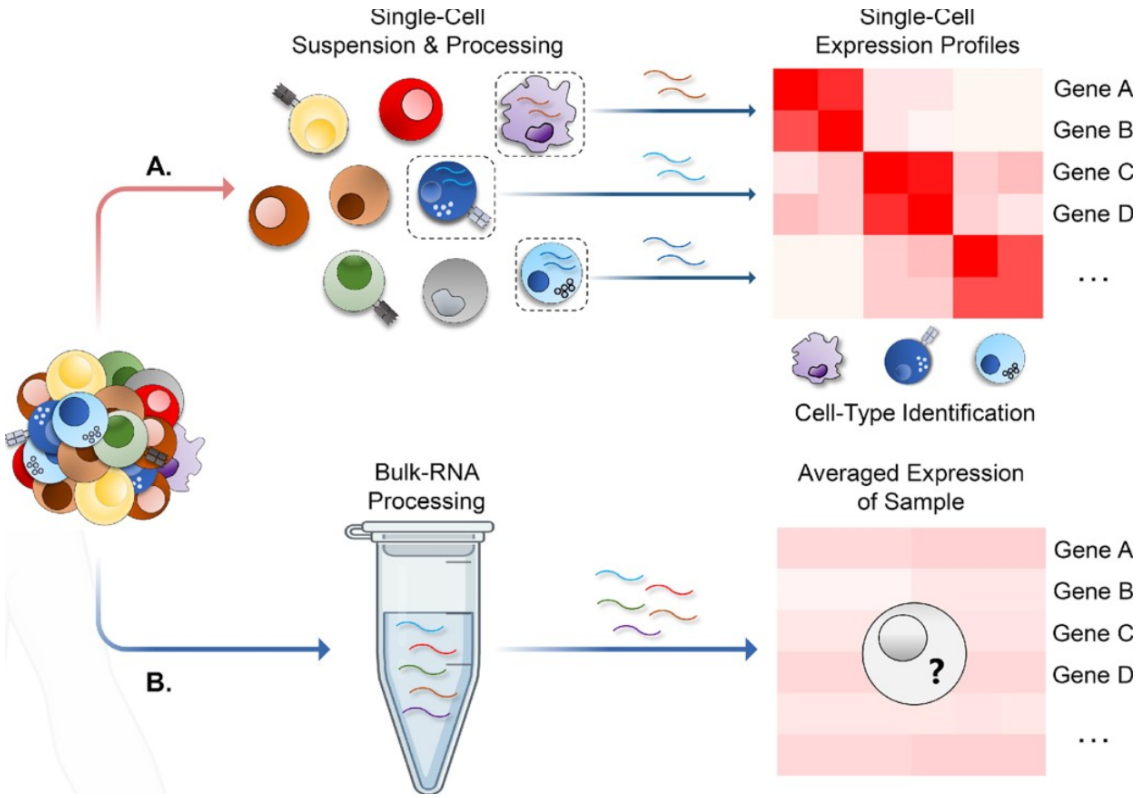


**Figure 2.6:** In bulk RNA-Seq (B), the output consists of averaged expression data compared across samples, while single-cell datasets (A) present expression data at the individual cell level, revealing how various cell types influence overall expression. Figure taken from [6]

## 2.2 Graph Structure and GRN Inference Goal

The following definition outlines the structure of a directed network, which is essential for understanding the graph-based representation of gene regulatory networks:

**Definition (Network):** A directed network (or graph) is defined as a pair $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a finite set of vertices (or nodes) and $\mathcal{E}$ represents the edges (or arcs) connecting these vertices. If $\mathcal{I}$ is the index set for the nodes, then the edges form a subset of the Cartesian product $\mathcal{I} \times \mathcal{I}$, where an element $(i, j)$ indicates the presence of an edge from node $i$ to node $j$. In contrast, an undirected network has a symmetric edge set, meaning that if an edge $(i, j)$ exists, the reverse edge $(j, i)$ must also be present [18].

The goal of gene regulatory network (GRN) reconstruction is to infer regulatory interactions from gene expression data, resulting in a directed graph where each node represents a gene, and each directed edge signifies a regulatory link from gene $i$ to gene $j$. These edges are unsigned, meaning that gene $i$ can act as either an activator or repressor of gene $j$ [11]. The GRN inference algorithms assign a confidence score to each edge, indicating the likelihood of a true regulatory relationship, with higher scores reflecting stronger evidence for regulatory influence. Network inference can be framed as a binary classification problem, where the existence of a regulatory link between genes is a positive instance, and the absence of regulation is a negative instance [29, 30]. Additionally, the number of established regulatory relationships between TFs and target genes is far smaller than the number of cases where no regulatory relationship exists. This disparity creates an imbalanced classification problem where positive instances are greatly outnumbered. It's important not to overlook this challenge, as ignoring it can result in high accuracy scores simply by categorizing most samples as negatives [31].

## 2.3 GRN Inference with Single-Cell Data

Early computational methods for gene regulatory network inference were designed around bulk sequencing technologies like microarrays and RNA-seq, which measured RNA expression across entire cell populations but were unable to capture the complexity of individual cells. With the rise of single-cell omics technologies, such as scRNA-seq, the ability to explore cellular heterogeneity at single-cell resolution has vastly improved. These advancements have fueled the development of new computational approaches that can now infer regulatory relationships at

the cell type, cell state, and single-cell level [7]. As discussed earlier, scRNA-seq data is notably noisier and more variable than bulk RNA-seq. The combination of technical noise and biological variation poses considerable challenges for computational analysis. Although a range of tools has been developed for bulk RNA-seq analysis, many of these cannot be directly applied to scRNA-seq due to its unique characteristics [26].

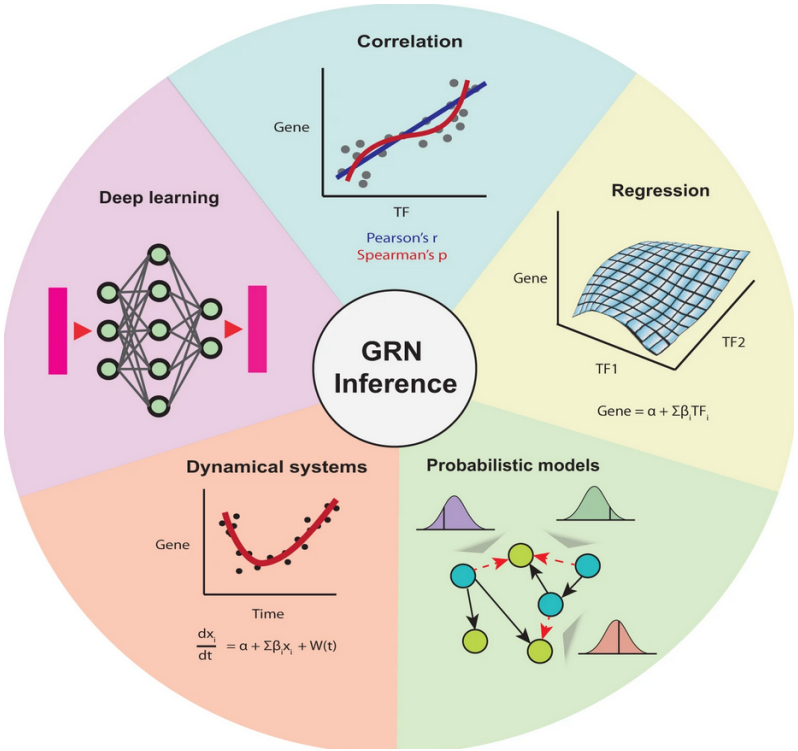


**Figure 2.7:** GRN Inference Methods: Correlation-based methods identify pairs of variables with similar variation patterns. Regression-based approaches predict gene expression using multiple predictors. Probabilistic models focus on finding the most likely regulators for a gene. Dynamical systems-based approaches model gene expression changes influenced by biological factors. Deep learning-based methods leverage neural networks to infer complex relationships among genes. Figure taken from [7]

GRN inference utilizes statistical and algorithmic methods to reveal the connections between genes and their regulators. By employing techniques like correlation, regression, probabilistic models, dynamical systems, and deep learning, researchers can accurately model and infer the regulatory frameworks that govern biological systems [7].

### 2.3.1 Correlation-Based Approaches

A widely used approach for reconstructing GRNs is based on the principle of "guilt by association," where co-expressed genes are presumed to be functionally related or co-regulated. Commonly applied association metrics include Pearson's correlation for detecting linear relationships and Spearman's correlation, a non-parametric alternative that can capture both linear and nonlinear associations [7]. Given two zero-mean vectors $v_i$ and $v_j$, the Pearson correlation between them is defined as:

$$\text{corr}(v_i, v_j) = \rho_{ij} = \frac{v_i \cdot v_j}{\|v_i\|\|v_j\|}$$

where $\cdot$ denotes the scalar (dot) product, and $\|v_i\|$ represents the Euclidean norm of vector $v_i$. In practice, given a set of $N$ expression measurements (e.g., under different conditions) for $p$ genes, these measurements are arranged into a data matrix $D$. Calculating the correlations between the columns of $D$ results in a $p \times p$ matrix of pairwise gene correlations, which can serve as the weights of an undirected network. By applying a suitable threshold, one can derive the network structure. Variations of this method involve using alternative correlation measures or applying a power transformation to the correlations to reduce noise from spurious low values [18]. Though correlation analysis can offer insights into potential regulatory relationships, it has limitations. It cannot determine the direction of regulation between two correlated transcription factors or account for regulation by a third factor. Additionally, correlation struggles to distinguish between direct and indirect relationships, especially in the presence of confounders [7].

### 2.3.2 Regression-Based Approaches

The dependence of two variables can also be assessed through an alternative approach that involves predicting one variable based on the other. The simplest way of achieving this is by using a linear regression approach, where the slope of the regression line determines the strength of the relationship between the variables. In the context of GRN, this involves regressing each gene on all other genes to determine network weights. For a gene $j$, where $x_k^j$ is its expression in sample $k$, this is done by solving the associated regression equation.

$$\mathbf{x}_k^j = \sum_{i \neq j} w_i \cdot \mathbf{x}_k^i + \varepsilon_k$$

where $\varepsilon_k$ is the noise term. The resulting weight $w_i$ serves as the weight for the network edge connecting gene $i$ to gene $j$. It is important to note that this regression formulation inherently assigns a direction to the network, although bidirectional edges are also possible. Regression models in GRN inference estimate the relationship between gene expression and multiple transcription factors or regulatory elements. Coefficients in the model represent the strength and direction of regulatory interactions. However, using many predictors can lead to overfitting and instability, especially when predictors are correlated. Non-parametric methods, such as tree-based regression, offer flexibility but are harder to interpret and more computationally intensive [18, 7]. To extend beyond the linear regression model for gene expression, a more general definition of gene regulation is introduced. In this broader framework, the expression of each gene under a given condition is still assumed to be influenced by the expression of other genes in the network, alongside random noise. Let $\mathbf{x}_k^{-j}$ represent the vector of expression values in the $k$-th experiment for all genes except gene $j$:

$$\mathbf{x}_k^{-j} = (x_k^1, \ldots, x_k^{(j-1)}, x_k^{(j+1)}, \ldots, x_k^p)^T,$$

The assumption is:

$$x_k^j = f_j(\mathbf{x}_k^{-j}) + \varepsilon_k, \quad \forall k,$$

Here, $\varepsilon_k$ is random noise with zero mean, conditional on $\mathbf{x}_k^{-j}$. The function $f_j$ represents the relationship between the expression of gene $j$ and the expressions of other genes in the network. This function can take various forms, including linear regression or more complex nonlinear models. It is further assumed that the function $f_j$ depends only on the expression of genes that are directly connected to gene $j$ in the targeted network. Identifying the regulatory links pointing to gene $j$ thus involves finding those genes whose expression predicts the expression of the target gene [11].

Among the advanced methods for modeling $f_j$, regression trees provide flexible approaches for capturing gene regulatory relationships. Classification and regression trees are machine-learning techniques that build prediction models by recursively partitioning the data space and fitting a simple predictive model within each partition. The partitioning process is represented as a decision tree. While classification trees are used for dependent variables that take on a finite number of categorical values, where the error is measured by the cost of misclassification, regression trees are applied to continuous or ordered discrete dependent variables, with the prediction error typically assessed by the squared differences between the observed and predicted values [32].

Regression-based methods are popular and scalable for reconstructing directed networks, but they are generally more computationally intensive than other data-driven approaches. They can predict gene expression levels based on a subset of genes and capture high-order conditional dependencies among gene expression patterns, unlike correlation-based methods that focus solely on pairwise dependencies. A significant challenge with regression models is that they can struggle to accurately identify relationships when data is limited. This is because many genes may have expression patterns that are highly correlated with each other, making it hard to determine which genes truly influence others [18].

### 2.3.3 Probabilistic Approaches

Probabilistic models for GRN inference use graphical models to estimate regulatory relationships by identifying the most probable connections based on the data. Probabilistic methods help to filter and rank regulatory interactions. By filtering out less relevant interactions, these models streamline the process, allowing researchers to focus on the most promising or significant relationships in their further studies. However, these models often rely on assumptions about gene expression distributions which may not be suitable for all genes [7]. Two prominent classes of probabilistic approaches are Gaussian Graphical Models and Bayesian Networks. GGMs treat gene expression data as a multivariate normal distribution, leveraging the precision matrix to identify partial correlations among genes. GGMs face challenges with high-dimensional data and the assumption of normality. On the other hand, Bayesian Networks construct a joint probabilistic model from local conditional dependencies, using directed acyclic graphs to represent relationships among genes. This approach effectively integrates prior knowledge and manages uncertainty, although it presents significant computational challenges in identifying network structures. Both models offer valuable insights into GRN inference, each with distinct methodologies and limitations [18].

### 2.3.4 Dynamical Systems-Based Approaches

Unlike regression and probabilistic methods that directly model relationships between variables, dynamical systems-based approaches aim to capture how systems evolve over time. For GRN inference, gene expression is estimated by considering factors such as transcription factor regulation, basal transcription, and stochastic variations over time, often represented by differential equations. These equations model changes in gene expression as a function of other genes' expression and environmental influences, providing a quantitative framework that

closely mirrors the biological system's behavior [3, 7]. Differential equations offer a well-established way to describe system dynamics by relating the rate of change of a variable to its value. In the context of GRNs, gene expression levels are modeled as variables, with interactions encoded in parameters. Commonly, linear and time-homogeneous models are employed, facilitating the inference of GRN structures through various methods, including regression techniques and Bayesian approaches. Differential equation models provide continuous-time semantics, potentially enhancing mechanistic interpretations and mitigating the effects of experimental design choices. However, they still face computational and identifiability challenges [18].

### 2.3.5 Neural Networks and Deep Learning-Based Approaches

A widely used approach for constructing gene regulatory networks is the neural network, which is modeled after the central nervous systems of animals. This method is highly adaptable, capable of recognizing input patterns and modeling various functional relationships and data structures. Among the neural network models, recurrent neural networks (RNNs) are particularly effective for gene regulatory network construction. RNNs excel at capturing the complex, nonlinear, and dynamic interactions between genes, offering advantages such as biological relevance, resistance to noise, the ability to incorporate feedback loops, and handling internal states throughout the process [33].

Deep learning models are a type of machine learning method that have attracted considerable interest in various fields, including bioinformatics. These models utilize artificial neural networks, which can be organized in different ways to accomplish a range of tasks. For instance, a multi-layer perceptron is capable of tackling problems that involve predicting outcomes, while an autoencoder can help reduce the number of dimensions in a dataset. Autoencoders are particularly useful because they can handle different types of inputs and learn the relationships between them, which may indicate possible regulatory connections. Despite their versatility, deep learning models have some drawbacks. They typically require large amounts of training data since they make few assumptions about the underlying patterns in the data. Additionally, these models can have many parameters that need to be estimated, demanding significant computational power. Another challenge is that deep learning models are often less interpretable than traditional statistical models. This means that the results they produce can be hard to understand, as the values assigned to the different variables usually lack clear meaning. Nonetheless, recent advancements, such as saliency methods, aim to improve interpretability by highlighting the key features in the model. These features can then be used to pinpoint potential

transcription factor regulators [7].

The issue with many supervised methods for GRN inference is their focus on analyzing gene pairs, which often overlooks the broader network context. Some studies highlight the importance of local subgraphs in uncovering valuable insights about the connections between genes. This is where graph representation models, such as graph neural networks, come into play. Graph models can represent complex relationships between transcription factors and genes, as well as their neighbors, rather than just focusing on two endpoints. Graph neural networks (GNNs) are an extension of neural networks that address graph-related tasks like node classification, link prediction, and graph classification. GNNs use an iterative process to share information among nodes. After a set number of iterations, each node is represented by a feature vector that aggregates information from its neighboring nodes within a specified distance. The overall graph representation is formed by pooling the feature vectors from all nodes. The degree of enhancement in the quality of predictions depends on the specific dataset, as aspects like complexity, noise, and the presence of clear patterns can all affect the outcomes. Unlike traditional data types like tables and images, graph data has unique characteristics that can be challenging to work with. The structure of graphs is non-Euclidean, meaning they don't follow the usual geometric rules, which makes standard metrics for measuring distance less effective. GNNs address this by considering node attributes, edge attributes, and the overall arrangement of the graph to create embeddings that reflect the graph's structure. These node embeddings capture information about both the structural features and the attributes of neighboring nodes [34, 35]. Here, a technique for creating these node embeddings, known as the graph convolutional network (GCN), is presented. In GCN the network is denoted as G = {V, $\xi$}, where $V \in \mathbb{R}^N$ represents the set of nodes and $\xi$ represents the set of edges. The primary objective of the graph encoder is to iteratively aggregate features from neighboring nodes to learn the features of each node $v_i$. The $l$-th layer of GCN can be defined as:

$$h_i^l = \text{AGGREGATE}(\{h_j^{l-1} : v_j \in E_i\})$$

Here, $h_j^{l-1}$ represents the features of node $v_j$ in the previous $(l-1)$th layer. $E_i$ denotes the first-hop neighbors of node $v_i$ within the network, encompassing node $v_i$ itself. The aggregator function in the graph encoder updates the feature of node $v_i$ in the $l$th layer by integrating the features of neighboring nodes. In GRN inference, GCNs are utilized to extract gene features by integrating first-order neighbor data, which constitutes a GCN layer. Initially, it is assumed that each node in the network is self-connected. This assumption facilitates the definition of a

normalized adjacency matrix, represented as $\tilde{A}$. The matrix $\tilde{A}$ is calculated as $\tilde{A} = D^{-\frac{1}{2}}AD^{\frac{1}{2}}$, where $A$ denotes the adjacency matrix of the graph. The presence of an edge between nodes $v_i$ and $v_j$ is indicated by the element $A_{ij}$. The diagonal matrix $D$ is defined such that each diagonal element $D_{ii}$ equals the sum of the corresponding row in $A$. GCN is used as an aggregation function to update the features of nodes. From the node feature matrix $H^{(l-1)}$ of layer $(l-1)$, the feature matrix $H^{(l)}$ at layer $l$ is derived based on the following formula:

$$H^{(l)} = \text{ReLU}(\tilde{A}H^{(l-1)}W^{(l-1)} + b^{(l-1)})$$

ReLU is the activation function, $H^{(l-1)}$ represents the model's output at layer $(l-1)$. $H^{(0)}$ corresponds to the input feature matrix X. $W^{(l-1)}$ represents the trainable weight matrix, while $b^{(l-1)}$ corresponds to bias vector. The output of the last layer of the GCN, referred to as $H$, is used as the final gene features. This matrix $H$ has dimensions $p \times d_1$, with $p$ denoting the number of genes and $d_1$ indicating the dimension of the gene features [13].

## 2.4 Performance Metrics

### 2.4.1 Metrics in Binary Classification

Classification problems can be categorized based on the number of classes involved. In binary classification, there are only two classes, while multiclass classification involves more than two. For binary classification, the two classes are typically labeled as P (positive) and N (negative), and an unknown sample is assigned to one of these categories. A classification model, trained during the learning phase, is employed to predict the true class of unseen samples. This model produces either discrete or continuous outputs. A discrete output provides the predicted class label, whereas a continuous output estimates the probability of the sample belonging to a particular class [9]. A confusion matrix is a type of contingency table that illustrates the discrepancies between the actual and predicted classes for a given set of labeled examples, as demonstrated in Figure 2.8 [36]. It serves as the foundation for calculating key binary classification metrics, including accuracy, precision, false positive rate (FPR), and true positive rate (TPR), which are crucial for evaluating model performance. The confusion matrix links ground truth to predicted labels at various thresholds, where a specific threshold score determines whether an instance is classified as positive. When the optimal threshold is unknown, performance curves that show changes across a range of thresholds are valuable. These curves enable evaluation of

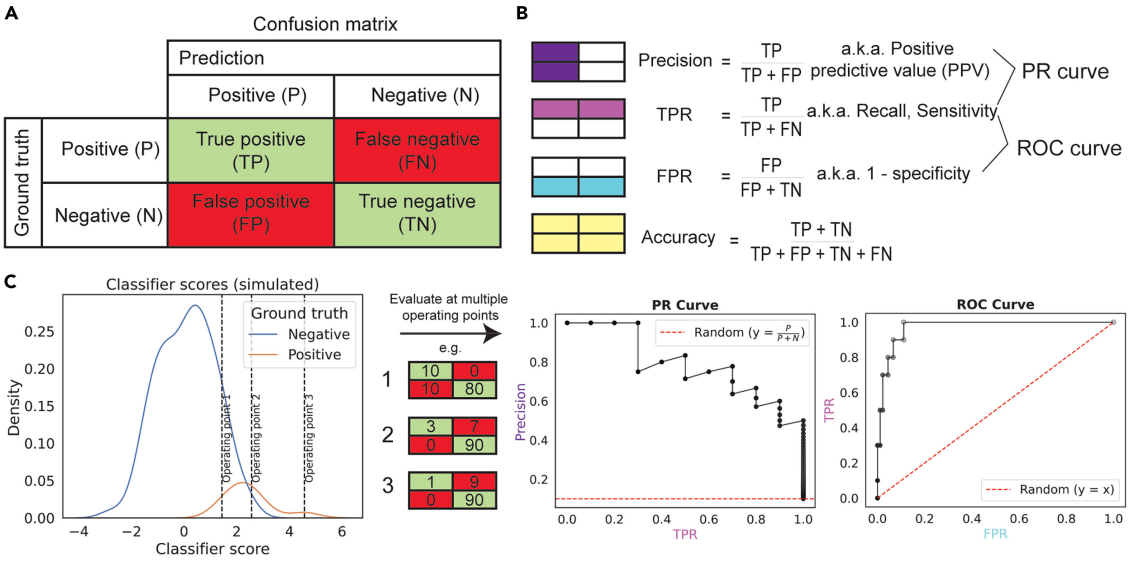


**Figure 2.8:** (A) A confusion matrix can be generated for a binary classifier at a specific threshold. (B) Metrics such as precision, true positive rate (TPR), and false positive rate (FPR) are derived from the confusion matrix. Precision and TPR are used in the precision-recall (PR) space, while TPR and FPR are used in the ROC space. (C) PR or ROC curves are generated by calculating these metrics at various thresholds, then interpolating the points and comparing the area under the curve across different classifiers. Figure taken from [8].

performance variations with different thresholds, and the area under the curve (AUC) summarizes the overall classification performance for comparison across different models [8].

In order to better understand the binary classification metrics derived from confusion matrix, it's important to first define its key components:

- **True Positive (TP):** The number of samples that are correctly identified as positive by the model.
- **True Negative (TN):** The number of samples that are correctly identified as negative by the model.
- **False Positive (FP):** The number of samples that are incorrectly classified as positive by the model.
- **False Negative (FN):** The number of samples that are incorrectly classified as negative by the model.

These four metrics serve as the basis for deriving various performance measures:

- **Accuracy (ACC)**: The proportion of correctly classified samples out of the total number of samples. Accuracy ranges from 0 to 1, with 1 indicating perfect classification and 0 indicating no correct predictions.

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}$$

- **Recall (REC)**: Also known as sensitivity or True Positive Rate (TPR), recall measures the proportion of actual positive samples that are correctly classified. It ranges from 0 to 1, where 1 indicates perfect identification of all positive cases, and 0 indicates that no positive cases are correctly predicted.

$$\text{REC} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- **Specificity (SPEC)**: The proportion of actual negative samples that are correctly classified. It is calculated as the ratio of correctly classified negative samples to all samples predicted as negative. Specificity ranges from 0 to 1, with 1 indicating perfect prediction of negatives and 0 indicating no correct predictions for the negative class.

$$\text{SPEC} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

- **False Positive Rate (FPR); 1 − Specificity**: The ratio of incorrectly predicted positive samples to the total number of actual negative samples.

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} = \frac{\text{FP}}{\text{N}}$$

- **Precision (PREC)**: The proportion of correctly identified positive samples among all samples predicted as positive. It is bounded between 0 and 1, where 1 signifies that all predicted positives are correct, and 0 indicates no correct predictions among the predicted positives.

$$\text{PREC} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

### 2.4.2 Imbalanced Datasets

Many biological problems involve binary classification tasks where instances of one class are greatly outnumbered by a much larger set of instances from the other class. This scenario, known as class imbalance, refers to datasets where instances are unevenly distributed, with some classes being significantly or even extremely more prevalent than others [8]. The minority class

is often the most important one to identify, but it's harder to do so. This is because the minority class might be linked to rare, important cases or because collecting data for these examples is expensive [37]. The substantial disparity in class distribution presents challenges for accurately identifying positive cases and for evaluating and ranking classifier performance. Effectively addressing class imbalance is therefore critical for various important tasks in biology [8].

In the case of balanced training data, most machine learning algorithms perform well. However, when dataset classes are imbalanced, these algorithms face challenges, often showing a bias toward the majority class. The inefficiency of these algorithms in dealing with imbalanced data stems from the fact that they aim to maximize performance measures like accuracy, which becomes less appropriate in such situations. Accuracy gives equal weight to correctly and incorrectly classified examples across different classes. For instance, in a dataset with 10% positive class and 90% negative class, a simple classifier that always predicts the majority (negative) class will achieve a high accuracy of 90%. However, this doesn't reflect the model's ability to detect the minority class. As data imbalance increases, more suitable metrics are needed to evaluate the classifier's performance, focusing more on the minority class and its distribution [38].

To address this, the receiver operating characteristic (ROC) curve is used. The ROC curve is a graphical representation used to evaluate the performance of classification models. This curve, originating from signal detection theory, plots the True Positive Rate (TPR) on the y-axis against the False Positive Rate (FPR) on the x-axis, providing a view of model performance independent of class imbalance. The ROC curve helps balance the trade-off between true positives and false positives. For classifiers with discrete outputs each classifier produces a single confusion matrix, which corresponds to one point on the ROC curve. To construct a complete ROC curve from such classifiers, methods like varying class proportions or using combinations of scoring and voting are utilized. In contrast, continuous output classifiers generate numeric scores representing the likelihood of a sample belonging to a specific class. By adjusting the threshold on these confidence scores, different points are obtained, collectively forming the ROC curve [9].

The ROC curve example highlights four key points:

- **Point (0,0)**: Represents a classifier with no positive classifications and correct classification of all negative samples (TPR = 0, FPR = 0).
- **Point (1,1)**: Represents a classifier where all positive samples are correctly classified, but all negative samples are misclassified.
- **Point (1,0)**: Indicates a classifier where both positive and negative samples are misclassified.

- **Point (0,1)**: Represents the ideal classifier, perfectly classifying all samples (perfect classification point).

The green curve, rising vertically from (0,0) to (0,1) and then horizontally to (1,1), illustrates perfect classification performance. Points in the ROC space above this line are better, while those below perform worse.

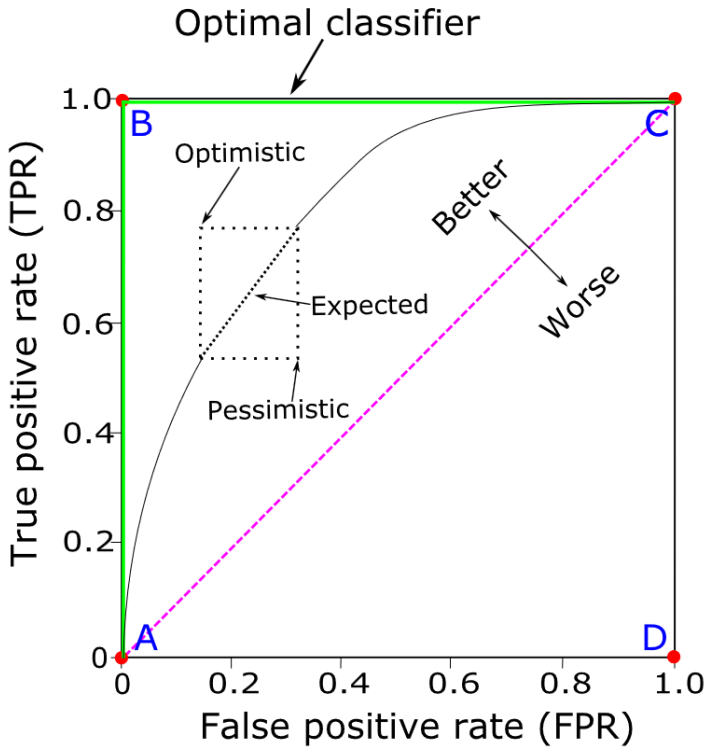


**Figure 2.9:** A ROC curve illustrating key points, along with the optimistic, pessimistic, and expected ROC segments for samples with identical scores. Figure taken from [9].

Due to the absence of a scalar value representing expected performance in the ROC curve, comparing different classifiers can be challenging. To facilitate this comparison, the Area Under the ROC Curve (AUROC) is used, providing a single value that quantifies the area under the ROC curve. The AUROC score ranges from 0 to 1, with a score of 0.5 indicating the performance of a random classifier. Figure 2.10 illustrates that classifier B, with a higher AUROC than A, generally performs better. The gray shaded area is shared by both classifiers, while the red area shows where B outperforms A. Despite B's higher AUROC, A performs better in the blue-shaded region. Note that classifiers with different ROC curves can have the same AUROC score [9].

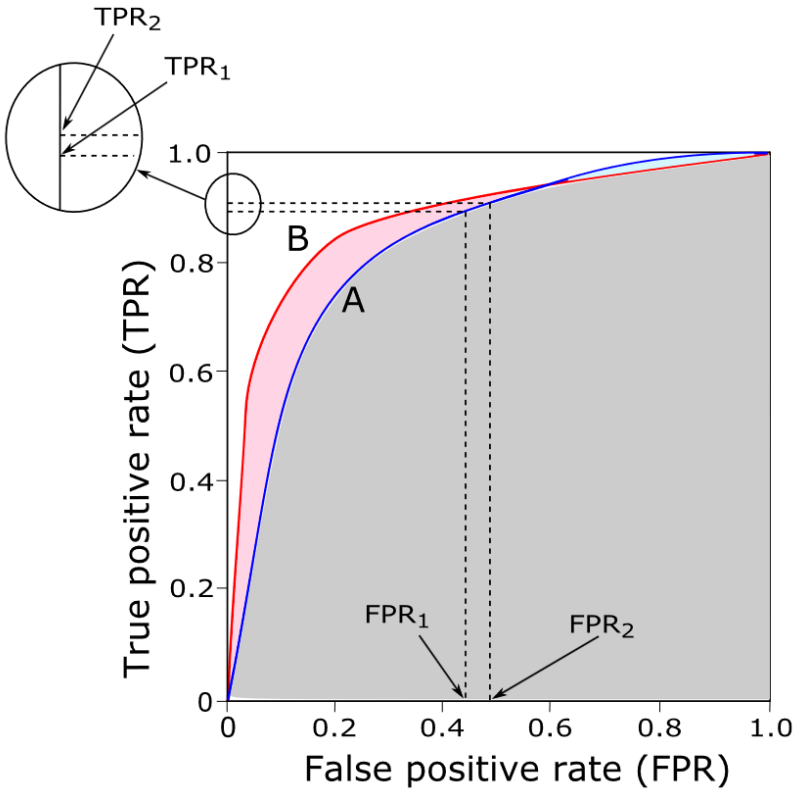


**Figure 2.10:** An example of AUROC metric. Figure taken from [9].

If there's a significant imbalance in the class distribution, ROC curves can present an overly optimistic representation of an algorithm's performance. In such cases, Precision-Recall (PR) curves are often recommended as a more suitable alternative to ROC curves [39]. The PR curve operates on a similar principle as the ROC curve and is generated by varying the classification threshold. However, while the ROC curve displays the relationship between sensitivity (or recall) and 1-specificity (FPR), the PR curve plots recall on the x-axis and precision on the y-axis. In essence, the x-axis of the ROC curve becomes the y-axis in the PR curve [9]. Precision and recall concentrate solely on positive examples and predictions, providing some insight into the rates and types of errors made. However, they do not offer any information about how effectively the model addresses negative cases. Recall is associated only with the positive examples, while precision is linked only to the positive predictions. Neither metric considers the number of true negatives [40]. Figure 2.11 illustrates that PR curves often have a zigzag shape, leading them to intersect more frequently than ROC curves. In a PR curve, the higher the curve, the better the classification performance. The ideal performance is depicted by the

green curve in Figure 2.11. This ideal PR curve begins at (0,1), indicating 100% precision and 0% recall, moves horizontally to (1,1) for perfect precision and recall, and then drops vertically to (1,0), where recall is perfect but precision is zero. The closer a PR curve is to the upper right corner, the better the model's performance. The endpoint of the PR curve can be calculated using the formula $(1, \frac{P}{P+N})$. This formula is important for two reasons. First, as the threshold value increases, recall also increases, eventually reaching its highest point at the endpoint. Second, raising the threshold affects both true positives (TP) and false positives (FP). When the dataset is balanced, meaning there are equal numbers of positive and negative examples, the precision at the endpoint is $\frac{P}{P+N} = \frac{1}{2}$. A horizontal line drawn at this precision level represents the performance of a random classifier. This line divides the PR curve into two regions: the area above the line represents good performance, while the area below the line indicates poor performance (as shown in Figure 2.11). The ratio of positive to negative examples in the dataset sets the baseline for this line. Therefore, if the ratio of positive and negative classes changes, it will shift this line and impact the overall classification performance [9].

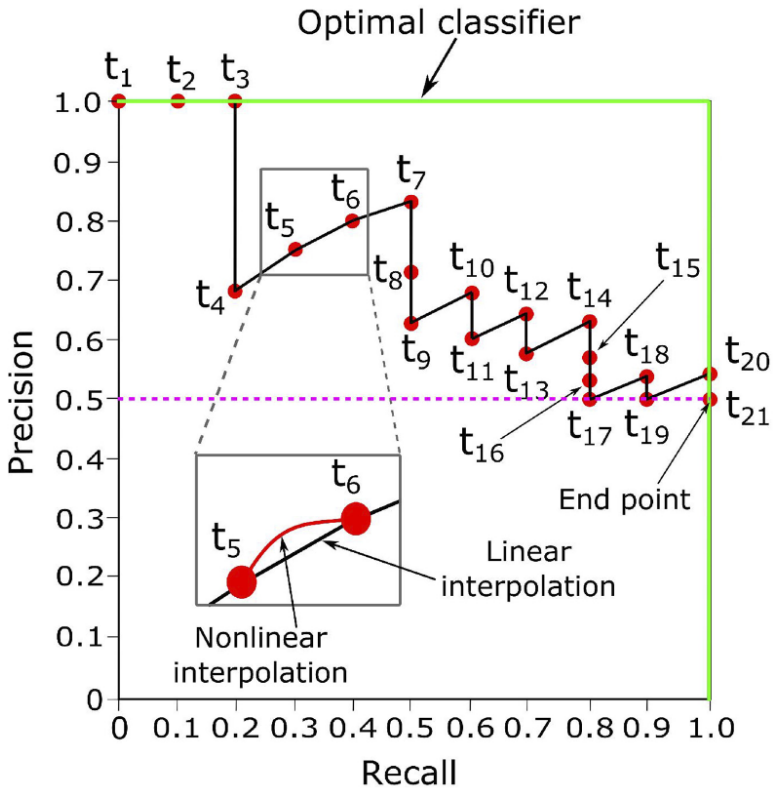


**Figure 2.11:** A PR curve example. Figure taken from [9].

It is often desirable to summarize the PR curve with a single scalar value, similar to how the AUROC is utilized for ROC curves. One common summary metric for the PR curve is the area under the PR curve (AUPRC) [41]. AUPRC ranges from zero to one, with random performance tied to the prevalence of positive examples in the dataset. The minimum possible AUPRC increases as prevalence rises because some parts of the precision-recall space become unattainable, even for the worst model. For instance, if a dataset has a prevalence of 0.5, a poor model that always predicts positives will still achieve a recall of 1 and a precision of 0.5, resulting in a minimum AUPRC of 0.31. Taking this baseline into account helps ensure more meaningful comparisons of performance across datasets with different prevalence rates [42].

# 3

# Datasets

One of the primary difficulties in evaluating gene regulatory network inference algorithms for single-cell RNA-seq (scRNA-seq) data is the absence of a known ”ground truth” network of regulatory interactions. Constructing a robust network is particularly challenging for higher organisms, such as vertebrates, due to the complexity and scale of their regulatory systems. Experimental approaches often require gain- and loss-of-function assays for individual regulators, as well as the identification of transcription factor binding sites. As a result, it is a common practice to generate artificial networks or to extract smaller subnetworks from extensive transcriptional networks to use as a reference. Most benchmarking efforts rely on simple model organisms like Escherichia coli and yeast, while for more complex organisms, ground-truth networks are typically limited to specific tissues or cell types and a small set of regulators. In research, the following three types of networks are often used as the ground truth for GRN inference. The first category consists of ”toy” networks with distinct topologies that generate various cellular trajectories with well-defined qualitative characteristics. The second category encompasses Boolean models that have been published and are used to study gene regulatory interactions involved in different developmental and tissue differentiation processes. The third category includes experimental scRNA-seq datasets and their corresponding ground-truth networks. [5, 10].

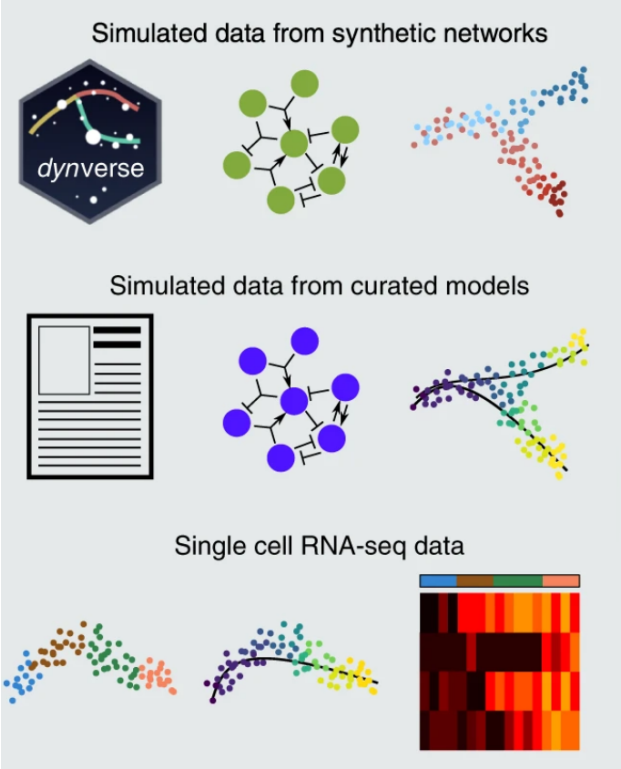


**Figure 3.1:** Overview of the three ground-truth network types used for evaluating GRN inference algorithms: synthetic toy networks, Boolean models, and experimental scRNA-seq networks. Figure taken from [10].

## 3.1 Datasets from Synthetic Networks

As previously mentioned, synthetic networks are often used to create datasets that simulate various gene expression patterns, providing a controlled environment for evaluating gene regulatory network inference algorithms. These networks are designed with different topologies, such as linear pathways, where genes follow a simple, sequential activation process; oscillatory circuits, where gene regulation occurs in repeating cycles; and bifurcating systems, where a single pathway splits at branching points, allowing cells to follow two distinct developmental or functional paths. The simulated datasets capture these dynamics, offering diverse temporal trajectories for evaluation. However, despite their usefulness, synthetic networks oversimplify biological complexity and may not fully capture the nuances of real gene regulatory systems. To address this, curated network datasets derived from published models of gene regulatory networks are often used [10].

## 3.2 Datasets from Curated Models

Curated network datasets derived from published models of gene regulatory networks (GRNs) provide a more accurate representation of biological control systems compared to synthetic networks. These datasets, based on Boolean models of GRNs, are particularly valuable for simulating gene regulatory processes involved in tissue differentiation and development—areas commonly studied through single-cell transcriptomic methods. Analyzing Boolean models from the literature enables insights into real biological network dynamics, offering a closer alignment with actual gene interactions. For instance, a model investigating ventralized spinal cord (VSC) development includes eight transcription factors connected by inhibitory interactions and accounts for five distinct neural progenitor cell types. Another model focused on hematopoietic stem cell (HSC) differentiation captures the transition of multipotent progenitor cells into distinct blood cell types, while gonadal differentiation (GSD) model illustrates the maturation of gonadal primordium into male or female gonads. These curated models allow for the simulation of steady states and cellular trajectories that align more closely with real single-cell data, grounding them in experimental findings and enhancing the understanding of regulatory processes in biology [10].

## 3.3 Experimental Single-Cell RNA-Seq Datasets

While curated network datasets derived from simplified Boolean models provide valuable insights into gene regulatory mechanisms, they often lack the complexity and biological realism needed for comprehensive analysis. To bridge this gap, experimental single-cell RNA-seq datasets are incorporated, capturing the dynamic expression profiles of genes across various cell types and developmental stages. These datasets, obtained from actual biological samples, offer a richer and more nuanced understanding of regulatory processes in living systems. Following text details the specific single-cell RNA-seq datasets used in this thesis, derived from both mouse and human samples. These datasets cover multiple cell types. Each dataset was processed following the procedures detailed in the corresponding publications. For datasets lacking normalized expression values, transcripts per kilobase million or fragments per kilobase million counts were log-transformed with a pseudocount of 1, and these values were used for analysis. Additionally, genes expressed in fewer than 10% of the cells were filtered out. Further details regarding the datasets and pseudotime computation are provided below [10].

**Mouse Embryonic Stem Cells (mESCs) [43]:** This dataset includes single-cell RNA se-

quencing (scRNA-seq) measurements for 421 primitive endoderm (PrE) cells that were derived from mouse embryonic stem cells. The data was collected at five different time points: 0, 12, 24, 48, and 72 hours. Pseudotime was calculated using the Slingshot method, with the cells measured at 0 hours as the starting point and those measured at 72 hours as the endpoint.

**Mouse Dendritic Cells (mDCs) [44]:** This dataset consists of over 1700 dendritic cells that were derived from bone marrow and subjected to various conditions. The study focused on wild-type cells stimulated with lipopolysaccharide, with measurements taken at 1, 2, 4, and 6 hours. Pseudotime was computed using Slingshot, starting with the cells measured at 1 hour and ending with those measured at 6 hours.

**Human hepatocyte-like cells (hHEPs) [45]:** This dataset originates from an scRNA-seq experiment involving induced pluripotent stem cells (iPSCs) that were cultured in two dimensions and differentiated into hepatocyte-like cells. It includes 425 scRNA-seq measurements collected at multiple time points: day 0 (iPSCs), and days 6, 8, 14, and 21 (mature hepatocyte-like cells). Pseudotime was calculated using Slingshot, starting with the cells measured on day 0 and ending with those measured on day 21.

**Human Embryonic Stem Cells (hESCs) [46]:** This dataset comes from a time course scRNA-seq experiment involving 758 cells undergoing differentiation to form definitive endoderm cells from human embryonic stem cells. Measurements were taken at 0, 12, 24, 36, 72, and 96 hours. Pseudotime was computed using the cells measured at 0 hours as the starting point and the cells measured at 96 hours as the endpoint.

After determining the pseudotime values for the cells in each dataset, the variation in gene expression across pseudotime was analyzed. The general additive model from the 'gam' R package was employed to calculate the variance and the corresponding p-value. To account for multiple hypothesis testing, the Bonferroni correction method was applied. The selection of genes for GRN inference began with all transcription factors (TFs) that had a variance P value of 0.01 or lower. To this set, an additional 500 and 1,000 genes were added. This method allowed for the inclusion of TFs with less pronounced variations in gene expression, which still play a regulatory role. Following the application of the GRN inference algorithm, only interactions originating from TFs were considered for further evaluation [10].

| Cell type | Cells | STRING | | | Non-specific ChIP-seq | | |
|---|---|---|---|---|---|---|---|
| | | TFs | Genes | Density | TFs | Genes | Density |
| hESC | 759 | 343(351) | 511(695) | 0.024(0.021) | 283(292) | 753(1138) | 0.016(0.014) |
| hHEP | 426 | 409(414) | 646(874) | 0.028(0.024) | 322(332) | 825(1217) | 0.015(0.013) |
| mDC | 384 | 264(273) | 479(664) | 0.038(0.032) | 250(254) | 634(969) | 0.019(0.016) |
| mESC | 422 | 495(499) | 638(785) | 0.024(0.021) | 516(522) | 890(1214) | 0.015(0.013) |

**Table 3.1:** Statistics for single-cell transcriptomic datasets and STRING and Non-specific ChIP-seq networks, including TFs and the 500 (1000) most variable genes

| Cell type | Cells | Cell-type-specific ChIP-seq | | | Loss Of Function/Gain Of Function | | |
|---|---|---|---|---|---|---|---|
| | | TFs | Genes | Density | TFs | Genes | Density |
| hESC | 759 | 34(34) | 815(1260) | 0.164(0.165) | - | - | - |
| hHEP | 426 | 30(31) | 874(1331) | 0.379(0.377) | - | - | - |
| mDC | 384 | 20(21) | 443(684) | 0.085(0.082) | - | - | - |
| mESC | 422 | 88(89) | 977(1385) | 0.345(0.347) | 34(34) | 774(1098) | 0.158(0.154) |

**Table 3.2:** Statistics for single-cell transcriptomic datasets and Cell-type-specific ChIP-seq and and Loss Of Function/Gain Of Function networks, including TFs and the 500 (1000) most variable genes

To evaluate the accuracy of a method for GRN inference, researchers compare the results to known interactions between transcription factors and their target genes found in public databases. These databases can contain different types of information. Some databases provide evidence that two genes (TF and its target) are related based on various links. These links can include: similar gene expression patterns, the genes evolving together over time, both genes being mentioned together in scientific papers. Other databases focus on specific interactions that have been confirmed through experiments. For instance, they collect data from ChIP-seq experiments that show where a TF binds to the DNA of target genes. Chromatin immunoprecipitation followed by sequencing (ChIP-seq) has become the standard method for mapping protein-binding locations and biochemical changes across the genome. It is a method for analyzing protein-DNA interactions by using antibodies to isolate specific proteins or DNA-bound nucleosomes. This technique allows for the precise mapping of these interactions across the genome, which was enhanced by the introduction of next-generation sequencing (NGS). Unlike its predecessor, ChIP-chip, which used microarrays to identify DNA-protein fragments,

ChIP-seq directly sequences the fragments, offering higher resolution, better coverage, and more accurate data [47, 48].

They also check if the DNA sequences where the TF binds have specific patterns called binding motifs. When researchers use single-cell RNA-seq data, they often validate inferred interactions using data from the same or similar cell types. In summary, different databases provide varying types of information about gene interactions, from broad functional links to specific experimental evidence, helping researchers assess the accuracy of their inference methods [49]. In this study, the following types of ground-truth datasets were utilized for experimental single-cell RNA-seq datasets:

- **Cell-type-specific:** For each experimental scRNA-seq dataset, the ENCODE [50], ChIP-Atlas [51], and ESCAPE [52] databases were searched for ChIP-seq data from the same or similar cell types.
- **LOF/GOF:** The loss-of-function/gain-of-function (LOF/GOF) dataset from the ESCAPE [52] database.
- **Nonspecific:** These networks include general transcriptional regulatory interactions not limited to specific cell types. Three key resources were used:

  **DoRothEA** [53]: Integrates ChIP-seq and transcriptional regulatory information from multiple sources. Two levels of evidence in this database were considered: A (curated/high confidence) and B (likely confidence).

  **RegNetwork** [54]: Includes genome-wide TF–TF, TF–gene, and TF–microRNA regulatory relationships in human and mouse collected from various sources. The TF–TF and TF–gene interactions were used for this analysis.

  **TRRUST** [55]: Contains TF–target interactions collected based on text-mining followed by manual curation for human and mouse.

- **Functional:** The human and mouse STRING [56] networks were used. An interaction here is functional and need not correspond to transcriptional regulation. This type of ground-truth network was selected due to the observation that many GRN methods predict indirect interactions for Boolean models [10].

This thesis exclusively utilizes experimental scRNA-seq datasets (hESC, mESC, hHEP, mDC), including those with transcription factors and 500 additional genes (TF+500), as well as those with transcription factors and 1000 additional genes (TF+1000), and their corresponding ground-truth networks (Cell-type-specific, Nonspecific, STRING and LOF/GOF), ensuring that the evaluation of GRN inference algorithms aligns with realistic biological conditions.

# 4 Methods

This chapter introduces the models evaluated in this study: GENIE3, a decision tree-based method, and scGeneRAI, which employs the explainable artificial intelligence technique of layerwise relevance propagation (LRP) for gene regulatory inference. These are unsupervised methods that infer regulatory interactions solely from expression data. In contrast, GNNLink and STGRNS are supervised methods that combine both expression data and known gene pairs to infer gene regulatory networks. GNNLink is a deep learning method focused on link prediction, while STGRNS leverages the transformer architecture. Following the introduction of the models, this chapter presents the contributions and hypotheses tested throughout the study. These contributions involve thorough evaluation of models' performance, modifications and extensions to the original models, as well as new experimental setups.

## 4.1 GENIE3

GENIE3 [11] (GEne Network Inference with Ensemble of Trees) is a decision tree-based method that gained recognition as the top performer in the DREAM4 In Silico Network Challenge [57]. It infers GRNs by using feature selection through ensembles of regression trees. Unlike linear regression models, tree-based methods like GENIE3 do not assume a specific nature of gene regulation, making them suitable for handling both combinatorial and non-linear interactions. Random forest regression, a notable tree-based approach, generates directed graphs of regulatory interactions, including feedback loops, resulting in more realistic GRNs [11].

The process begins by collecting a set of measurements from experiments, referred to as a learning sample. This sample consists of $N$ measurements, where each measurement contains the expression values of $p$ genes from a particular experiment. Learning sample is defined as follows:

$$LS = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\},$$

with $\mathbf{x}_k \in \mathbb{R}^p, k = 1, \ldots, N$ representing a vector of expression values of all $p$ genes in the $k$-th experiment:

$$\mathbf{x}_k = (x_k^1, x_k^2, \ldots, x_k^p)^T.$$

Using this learning sample, algorithm aims to predict the regulatory links between genes. Most algorithms begin by ranking the potential regulatory links from most to least significant. By setting a threshold on this ranking it is possible to obtain a practical network prediction. The issue of determining the optimal confidence threshold is not addressed in this task. Here, network inference algorithm is described as a procedure that uses a learning sample to assign weights $w_{i,j} \geq 0$, $(i,j = 1, \ldots, p)$ to potential regulatory links from gene $i$ to gene $j$. The goal is to produce higher weights for links that represent actual regulatory interactions [11].

GENIE3 approach involves breaking down the task of recovering a network of $p$ genes into $p$ subproblems, with each subproblem focused on identifying the regulators of a single gene. Using gene expression data, the aim is to find the subset of genes that directly influence or predict the expression of a target gene. This task is treated as a feature selection problem within supervised learning. Tree-based ensemble methods are used to rank features, and the process leverages this mechanism to identify regulatory genes. In the following paragraphs, the procedure for addressing the network inference problem through feature selection techniques will be outlined, followed by an application of the procedure using tree-based ensembles [11].

The expression of each gene in a given condition is modeled as a function of the expression values of other genes in the network, along with some random noise. The function that governs gene $j$'s expression is assumed to depend only on genes that are directly connected to gene $j$ in the network. Therefore, identifying regulatory links for gene $j$ involves determining which genes' expressions predict the expression of gene $j$. This problem can be viewed as a feature selection task in regression, with numerous existing solutions. In this context, a feature ranking method is employed, which orders the features based on their relevance to the output, instead of directly providing a subset of features [11].

The network inference procedure (shown in Figure 4.1) is carried out as follows. For each gene, the algorithm generates a learning sample consisting of input-output pairs, where the

input includes the expression values of all genes except the target gene. Feature selection techniques are then applied to determine the strength of the relationship between the target gene and each of the other genes. These relationships are represented by confidence scores that reflect the likelihood of a regulatory connection. Finally, the confidence scores for all genes are aggregated to produce a global ranking of the regulatory links, providing a comprehensive view of the gene network. The problem's complexity and the proposed solution impose constraints on feature selection techniques. These constraints include the expectation that functions $f_j$ have to involve multiple genes and exhibit non-linearity, while the number of input features exceeds the number of observations. Efficient computation is crucial as the algorithm needs to be rerun $p$ times for $p$ genes, emphasizing the need for speed and minimal manual tuning. Tree-based ensemble methods are well-suited for the task because they do not assume the nature of the target function. They can handle interactions between features and non-linear relationships effectively. Additionally, these methods are efficient in scenarios with numerous features, offering fast computation, scalability, and requiring minimal parameter tuning [11].

Each subproblem, represented by a learning sample $LS^j$, is approached as a supervised regression task. The goal is to minimize prediction errors using square error loss. This involves finding a function $f_j$ that accurately predicts the expression of gene $j$ based on the expressions of other genes in the network. Regression trees are employed to tackle this challenge by creating hierarchical models. The core principle of this approach is to recursively divide the learning sample using binary tests on selected input variables, aiming to reduce the variance of the predicted gene expression $x^j$ across different subsets of samples. For numerical variables, potential splits compare input variable values against a dynamically determined threshold during the tree-building process. This method effectively addresses complex relationships and non-linearities in gene expression data, facilitating accurate predictions through iterative refinement of predictive models [11].

Ensemble methods often enhance the performance by combining predictions from multiple trees. In the presented network inference procedure Random Forests and Extra-Trees are evaluated. These methods utilize randomization techniques to improve predictive accuracy and robustness in gene regulatory network inference. Each tree in the Random Forest ensemble is trained on a bootstrap sample from the original learning dataset. At each test node of the decision tree, a subset of $K$ attributes (features) is randomly selected from all available attributes. The best split at each node is determined based on these randomly selected attributes. Unlike Random Forests, each tree in the Extra-Trees ensemble is built on the original learning sample without bootstrap sampling. At each node of the tree, instead of evaluating all possi-

ble splits, Extra-Trees consider $K$ random splits. Each random split is determined by selecting one input feature (attribute) randomly and a threshold for that feature. For these two methods, ensembles of 1000 trees are grown, and two values of the main parameter are considered: $K = \sqrt{p-1}$ and $K = p - 1$, where $p - 1$ is the number of inputs, equal to the number of potential regulators of each gene [11].

One notable feature of tree-based methods is their ability to compute a variable importance measure from a tree, which ranks the input features based on their relevance in predicting the output. Out of various variable importance measures that have been proposed, the measure considered here computes the total reduction in the variance of the output variable at each test node $N$, as defined by:

$$I(\mathcal{N}) = \#S\mathrm{Var}(S) - \#S_t\mathrm{Var}(S_t) - \#S_f\mathrm{Var}(S_f)$$

$S$ represents the samples at node $\mathcal{N}$, $S_t$ and $S_f$ are subsets where the test is true and false, respectively, $\mathrm{Var}(.)$ is the variance of the output in a subset, and $\#$ indicates the number of samples in a set. The sum of $I$ values of all tree nodes where this variable is used for splitting determines the overall importance of one variable for a single tree. Variables not selected get a zero importance score, while those chosen near the root receive higher scores. For ensembles, attribute importance can be averaged across all trees, enhancing reliability due to variance reduction [11].

Each tree-based model generates an individual ranking of genes as potential regulators of a target gene, with weights $w_{i,j}$ calculated as the sums of total variance reductions as defined previously. The total variance of the output variable explained by the tree can be equated to the sum of the importances of all variables in the tree. For unpruned trees, such as those used in Random Forests and Extra-Trees ensembles, this value typically closely approximates the initial total variance of the output variable

$$\sum_{i=j} w_{i,j} \approx N\mathrm{Var}(S)$$

S represents the learning sample used to build the tree, and $\mathrm{Var}(S)$ denotes the variance of the target gene estimated within that specific learning sample. To mitigate bias in regulatory link ordering based on weights $w_{i,j}$, which could favor highly variable genes, initially gene expressions are normalized to have unit variance across the training set. This normalization step precedes the application of tree-based ensemble methods [11].

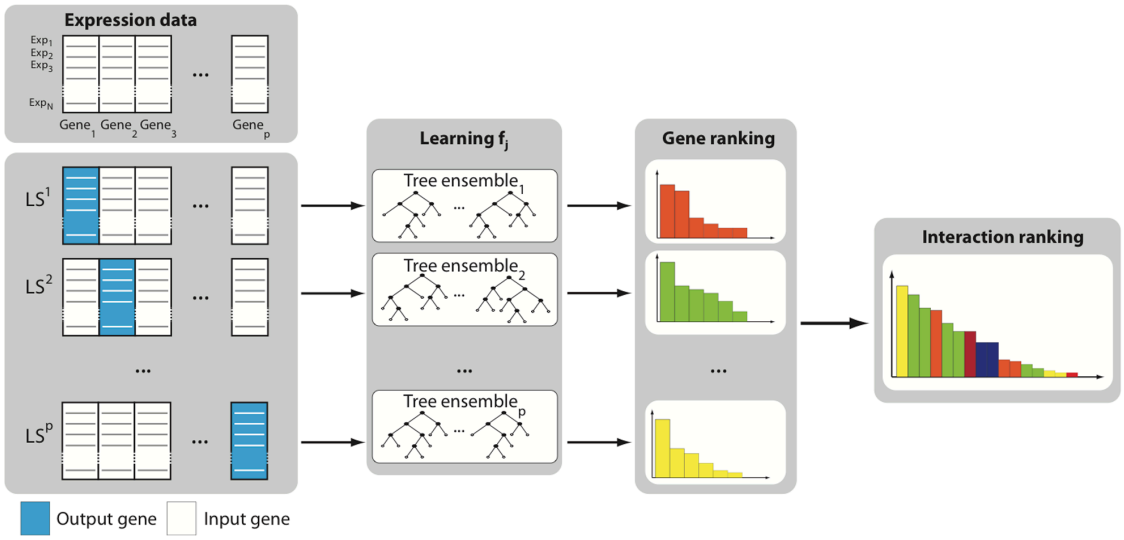


**Figure 4.1:** The GENIE3 procedure generates a learning sample ($LS_j$) for each gene $j \in \{1, \ldots, p\}$, with gene $j$'s expression levels as output and all other genes' expression levels as inputs. A function $f_j$ is learned from $LS_j$, and a local ranking of all genes excluding $j$ is computed. Global ranking of regulatory links is formed by aggregating the $p$ local rankings. Figure taken from [11]

## 4.2 ScGeneRAI

The scGeneRAI [12] (single-cell Gene Regulatory network prediction by explainable AI) approach to the problem of gene regulatory inference is centered on employing the explainable artificial intelligence method layerwise relevance propagation (LRP). To predict single-cell gene regulatory networks, scGeneRAI works in two main steps: First, it trains a deep neural network to estimate the expression level of a specific gene. It does this by looking at random sets of other genes. Essentially, it learns how the expression of one gene can be influenced by the expression of other genes. After the neural network is trained, LRP is used. LRP helps determine how important each of the other genes is in making the prediction about the specific gene. It analyzes the neural network's decisions to figure out which genes contribute most to the prediction. The input dataset consists of $N$ samples (cells) and $p$ features (genes). Initially, a neural network is trained to estimate genes in a sample using a random subset of other genes. Then, a gene prediction is performed based on K randomly selected sets of other genes, using the trained neural network. In order to determine the relevance of each gene for the prediction of the target gene, LRP is applied after each prediction. Finally, by averaging these relevance over all K repetitions, raw LRP values between target gene and the predicting genes are gen-

erated. After repeating this procedure for every possible target gene a full matrix of raw LRP values with dimension $p \times p$ is created [12].

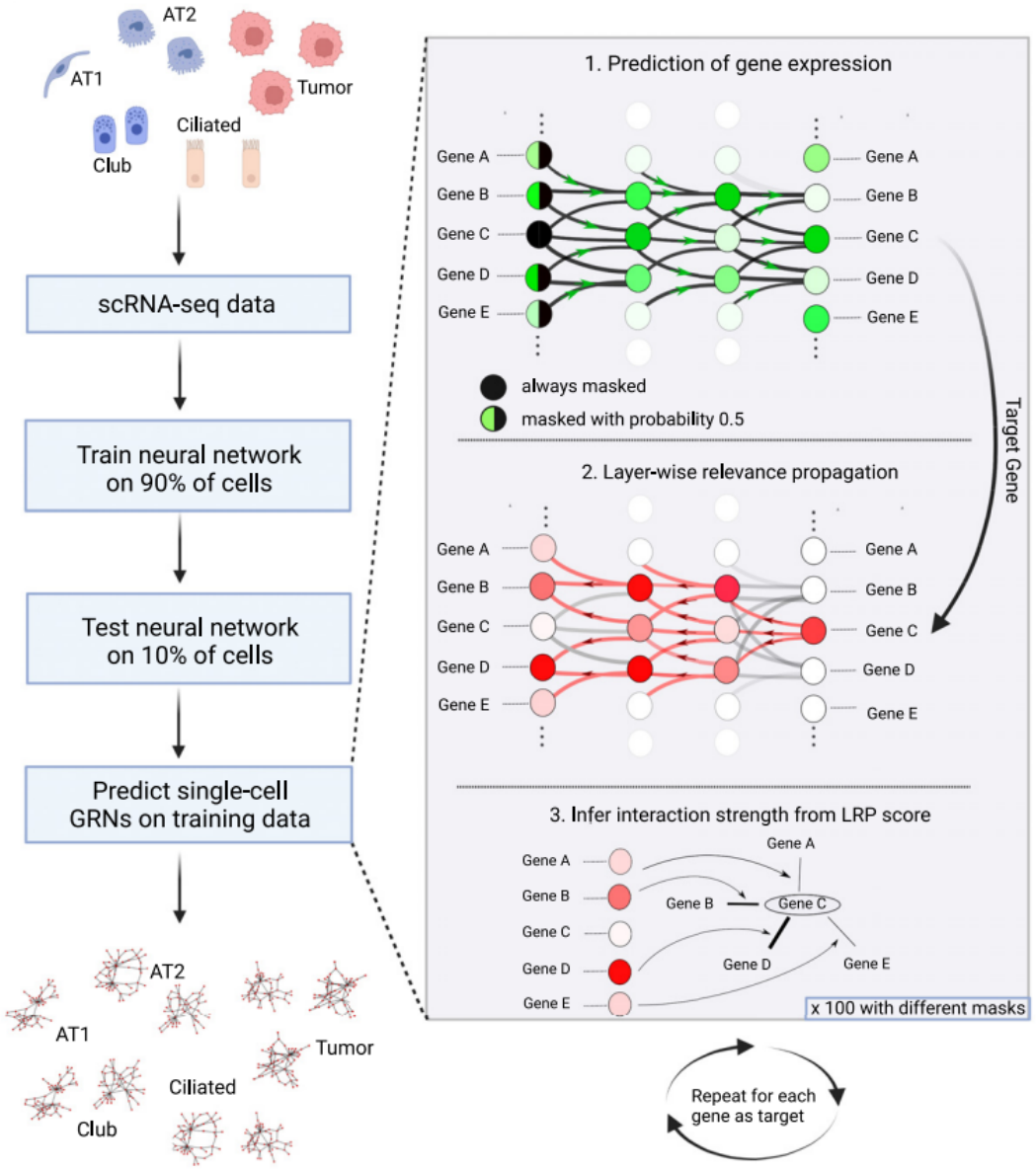


**Figure 4.2:** Workflow for inferring single-cell GRNs using scGeneRAI: A neural network is trained on scRNA-seq data to predict the expression of each gene based on selected sets of other genes. After training, the single-cell GRN is predicted in three steps: (1) Predict the target gene's expression using a set of predictor genes. (2) Use LRP to assess the relevance of each gene in the prediction. (3) The LRP scores are then used to quantify the interaction strength between the target gene and all predictor genes. This process is repeated for $n$ masks and for all genes as target genes. Figure taken from [12].

The neural network consists of two hidden layers, each with a width proportional to the

number of genes (10·p), allowing its capacity to scale with the dataset. It is designed to impute missing gene data, where p is the total number of genes, and q is the number of missing genes. Each gene with an abundance $a$ is encoded as a tuple $(a, 1-a)$, with $(0,0)$ representing missing genes. The input vector is thus of size $2 \cdot p$ and maps to an output vector of size $p$. Training parameters include stochastic gradient descent with a learning rate of 0.02 and a PyTorch learning rate scheduler with a weak exponential decay ($\gamma = 0.995$) to ensure network convergence. The batch size is set to 5, and momentum was configured at 0.9. The log hyperbolic cosine loss function is utilized as the training loss. Using the trained neural network model, it is possible to identify which input genes contribute to the prediction of each output gene. The function $F$ of the neural network takes input genes to produce output genes and is defined as: $(y_1, \ldots, y_p) = F(x_1, x_2 \ldots, x_p,)$. The goal is to compute a matrix of relevance scores $R_{il}$ that shows how much each input gene $i$ contributes to each output gene $l$. This process is known as attribution. Given its robustness and computational efficiency, the Layer-wise Relevance Propagation is utilized. In a single forward/backward pass through the network Layer-wise Relevance Propagation allows extraction of a collection of scores $(R_{il})_{i=1}^{p}$ for each output $y_l$. LRP initiates from the neural network's output, beginning with a specific predicted gene value $y_l$. It then iteratively propagates this score through the network layers towards the input, employing a systematic layer-wise approach. The activation of neuron $k$ is defined by the equation

$$a_k = \rho\left(\sum_{0,j} a_j w_{jk} + b_k\right)$$

with $j$ and $k$ representing indices for neurons in two adjacent layers, while $a_j$ and $a_k$ represent their respective activations. $w_{jk}$ and $b_k$ are parameters learned from data, $\rho$ is either a ReLU or linear activation function, and $\sum_{0,j}$ sums over all input neurons $j$ plus a bias (represented by a constant activation $a_0 = 1$ and weight $w_{0k} = b_k$). The following propagation rule, known as 'generalized LRP-$\gamma$' is used to propagate relevance scores to a lower layer (i.e. from the layer of neouron $k$ onto the layer of neuron $j$):

$$\begin{aligned} R_j = &\sum_k \frac{a_j^+ \cdot (w_{jk} + \gamma w_{jk}^+) + a_j^- \cdot (w_{jk} + \gamma w_{jk}^-)}{\sum_{0,j} a_j^+ \cdot (w_{jk} + \gamma w_{jk}^+) + a_j^- \cdot (w_{jk} + \gamma w_{jk}^-)} \cdot 1_{a_k > 0} \cdot R_k \\ &+ \sum_k \frac{a_j^+ \cdot (w_{jk} + \gamma w_{jk}^-) + a_j^- \cdot (w_{jk} + \gamma w_{jk}^+)}{\sum_{0,j} a_j^+ \cdot (w_{jk} + \gamma w_{jk}^-) + a_j^- \cdot (w_{jk} + \gamma w_{jk}^+)} \cdot 1_{a_k < 0} \cdot R_k \end{aligned}$$

Here, $R_k$ represents the relevance score assigned to neuron $k$ as $y_l$ propagates backward from the top layer to neuron $k$'s layer in the neural network. $(\cdot)^+$ and $(\cdot)^-$ denote $\max(0,\cdot)$ and $\min(0,\cdot)$, respectively. Explanation quality is maximized by appropriately selecting the hyperparameter $\gamma$. Finally, at the input layer, there are $2p$ explanation scores that indicate gene contributions, with each gene represented by a pair of values. The desired $p$ relevance scores are obtained by summing the remaining $2p$ scores into an $p$-dimensional vector that represents each gene's contribution. Repetition of the LRP procedure $K$ times for random sets of predicting genes yields the raw LRP score ($\text{LRP}_r$) which represents the average over these sets. Afterwards, $\text{LRP}_r$ scores are calculated for all predicted genes, producing an $p \times p$ matrix that represents gene-to-gene interactions [12].

## 4.3 GNNLink

GNNLink [13] is a supervised deep learning method that aims to infer gene regulatory networks through link prediction. The GNNLink framework (shown in Figure 4.3) consists of several key components: raw data preprocessing, interaction graph encoder based on GCN and GRN reconstruction. The input for GNNLink consists of scRNA-seq gene expression data and a generic GRN (known gene pairs) [13].

After preprocessing raw single-cell expression data, a GCN-based interaction graph encoder is used to learn gene features by leveraging the structure of the gene interaction graph. This allows the model to gather information from nearby nodes and develop features that capture the structure of the surrounding network. These obtained features, represented as a gene feature matrix $H$, are then used as the foundation for predicting gene regulatory dependencies. The GNNLink model computes dot products on gene feature vectors to derive comprehensive gene regulatory relationships:

$$R = \text{ReLU}(HH^T)$$

Here, ReLU represents the activation function. The matrix $R$ represents reconstructed GRNs, with each element denoting the regulatory action score for a gene pair.Expanding on this outlined method for computing the score matrix $R$ of the reconstructed gene regulatory network, the loss function is defined as:

$$\ell = \sum_{(i,j)\in\Omega^+\cup\Omega^-} \Phi(R(i,j), A(i,j)) + \lambda\|\Theta\|_F^2$$

Here, $\Omega^+$ and $\Omega^-$ represent the positive and negative sample sets used in model training, respectively. The parameter matrix $\Theta$ of the GNNLink model is denoted by $\Theta$, with $\lambda$ serving as a weight factor to adjust its impact. The optimization process employs mean square error (MSE) loss $\Phi(.)$, quantifying the difference between predicted gene regulation scores and actual dependencies in the datasets. The Adam optimizer iteratively updates model parameters until convergence, enabling the derivation of the gene regulation score matrix $R$. Higher values of $R_{ij}$ indicate stronger regulatory connections between genes $i$ and $j$. During training, encoder parameters are simultaneously updated with gene features [13].

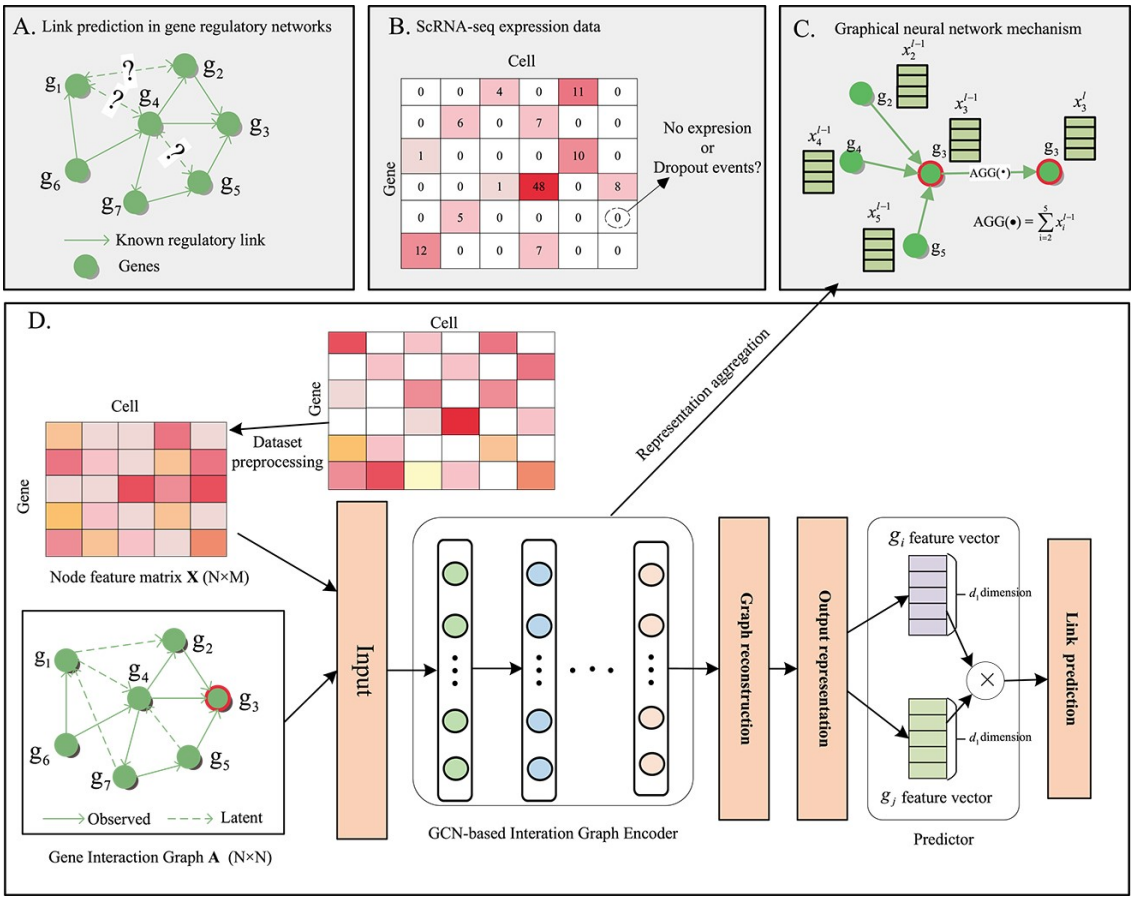


**Figure 4.3:** Overview of the GNNLink framework: (A) GRN inference is framed as a linkage prediction problem, aiming to identify potential edges based on existing ones. (B) Imputation of scRNA-seq expression data. (C) Learning node features, where AGG(·) aggregates features from connected nodes, such as node 3. (D) The GNNLink model consists of three main steps: preprocessing raw data, learning node features to capture key gene information, and reconstructing the interaction graph for link prediction, with gene interdependencies represented by the dot product. Figure taken from [13].

## 4.4 STGRNS

STGRNS [14] is a supervised method that utilizes the transformer architecture. Key modules that comprise the structure of STGRNS include the GEM (Gene Expression Motif) module, the positional encoding layer, the transformer encoder, and the classification layer. The purpose of the GEM module is to reconfigure gene pairs into a format suitable for input by the transformer encoder. The positional encoding layer captures positional or temporal information, while the transformer encoder calculates the correlation of different sub-vectors. The final classification output is produced by the classification layer [14].

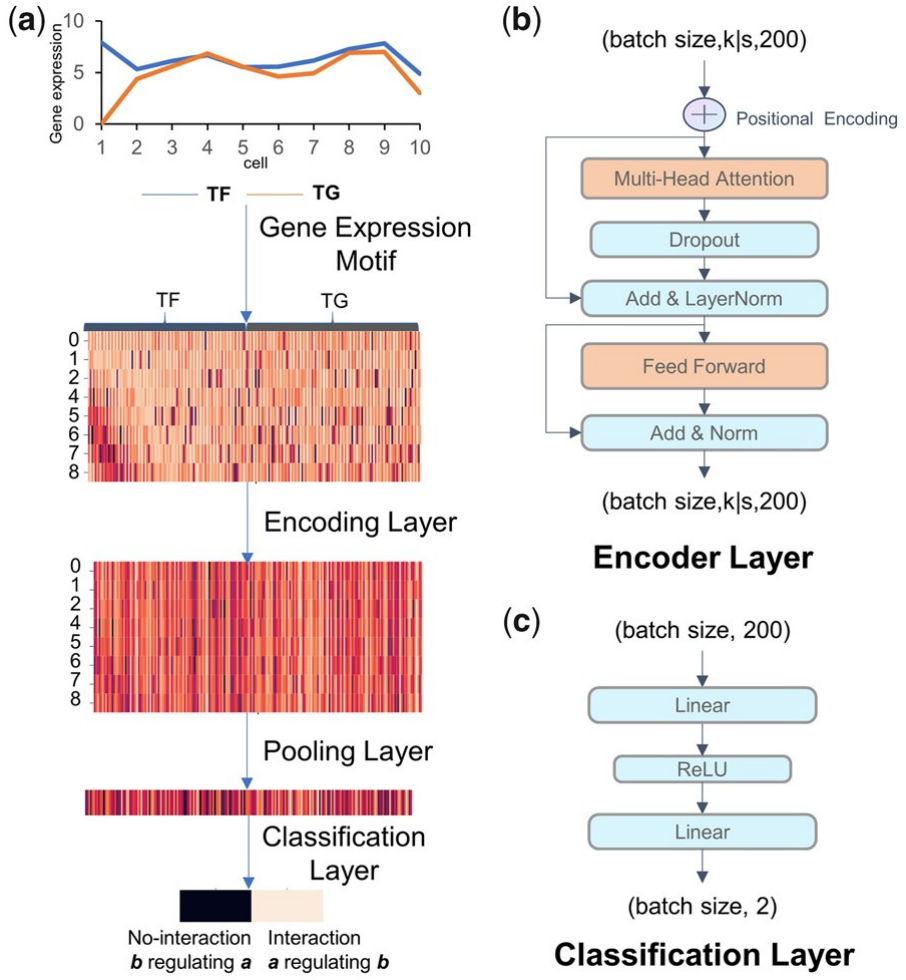


**Figure 4.4:** STGRNS architecture: (a) The processing flow of each gene pair; (b) Encoder layer and (c) classification layer; Figure taken from [14]

GEM is a data processing approach based on the assumption that the expression values of genes regulated by a shared transcription factor (TF) are synchronous for certain spans or phases. $X_i = (X_{i,1}, X_{i,2}, \ldots, X_{i,N})$ represents the expression vector of gene $i$, where $N$ is the number of cells. Contiguous subvectors of are generated from the vector $X_i$, with their length $s$ determined by a "window size" parameter. For gene $i$, the $l$-th sub-vector $X_{i,l}$ is defined as $X_{i,l} = (X_{i,ls+0}, X_{i,ls+1}, \ldots, X_{i,ls+s-1})$, where $l$ ranges from 1 to $\frac{N}{s}$. Following this segmentation, gene $i$ (represented by $X_i$) and gene $j$ (represented by $X_j$) in a gene pair $(X_i, X_j)$ are vectorized as $X_i = (X_{i,0}, X_{i,1}, \ldots, X_{i,\frac{N}{s}})$ and $X_j = (X_{j,0}, X_{j,1}, \ldots, X_{j,\frac{N}{s}})$, respectively. Each sub-vector in $X_i$ and $X_j$ is concatenated into a unified sub-vector $X_{ij,m}$, where $m$ varies from 0 to $\frac{N}{s}$. Formally, $X_i$ and $X_j$ are combined to form a new vector $X_{ij} = (X_{ij,0}, X_{ij,1}, \ldots, X_{ij,\frac{N}{s}})$, serving as the input for the transformer encoder [14].

When $Xij$ is put into the transformer encoder, the order or temporal information of gene expression vectors is lost. To retain positional information, sinusoidal positional encoding is used. Odd-numbered sub-vectors are represented by the sine function, and even-numbered sub-vectors by the cosine function, defined as follows:

$$PE(m, 2n) = \sin\left(\frac{m}{10,000^{2n/s}}\right) \quad \text{(Equation 1)}$$

$$PE(m, 2n+1) = \cos\left(\frac{m}{10,000^{(2n+1)/s}}\right) \quad \text{(Equation 2)}$$

Here, $m$ denotes the position $X_{ij,m}$ within $Xij$, $2n$ represents even sub-vectors, and $2n + 1$ represents odd sub-vectors.

Transformer encoder layer consists of two sub-networks: a multi-head attention network and a feed-forward network. The exceptional performance of the attention mechanism is largely due to several unique properties. One key feature is its ability to focus intensely on important sub-vectors within gene expression vectors. This is consistent with the proposed GEM. Thanks to this mechanism, STGRNS can disregard the negative impacts of insignificant sub-vectors. Another feature is its ability to capture connections globally, allowing it to fully utilize discontinuous sub-vectors to enhance the accuracy of STGRNS. Specifically, STGRNS utilizes the Scaled Dot-Product Attention mechanism. This process involves calculating a weighted sum of sub-vectors using weights determined by a softmax function. These weights are derived from the similarity (measured by dot-products) between Query (Q) and Key (K) vectors, ensuring that more relevant sub-vectors receive higher weights. This process helps STGRNS effectively capture and utilize relevant information from gene pairs during its computations. Mathemati-

cally expressed, the attention mechanism is:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{S}}\right)$$

where $Q = W^q X_{\text{posi}}$, $K = W^k X_{\text{posi}}$, $V = W^v X_{\text{posi}}$, the $W^{q,k,v}$ represents the linear project weight, $\text{softmax}(z_i) = \exp\left(\frac{z_i}{\sum_j z_j}\right)$. Multi-head self-attention is employed to capture diverse interaction information across multiple projection spaces. Setting the hyperparameter head to 2 in STGRNS enables two simultaneous self-attention operations.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}1, \text{head}2)$$

with $\text{head}_h = \text{Attention}(Q_h, K_h, V_h)$.

After multi-head attention is applied to $X_{\text{posi}}$, the resulting vectors undergo further processing through a residual connection and layer normalization, yielding

$$X_{\text{attention}} = \text{LayerNorm}(X_{\text{posi}} + X_{\text{attention}})$$

This step ensures that the outputs are stabilized and standardized before being passed as input to the feed-forward network. The residual connection preserves information from the original input $X_{\text{posi}}$, while layer normalization adjusts the scale and distribution of the vectors, optimizing them for subsequent processing in the network. This transformation prepares $X_{\text{attention}}$ to effectively capture and integrate features relevant to the task at hand within the feed-forward network. Self-attention, while capable of using adaptive weights and focus on all sub-vectors, may still miss capturing certain nonlinear features. To address this, a feed-forward network is used to enhances nonlinearity. This network consists of two linear layers with ReLU activation:

$$X_{\text{encoder}} = \max(0, X_{\text{attention}} W1 + b1) W2 + b2$$

STGRNS employs an average pooling layer to compute $X_{\text{average}} = \text{mean}(X_{\text{encoder}})$, consolidating encoded vectors. The subsequent classification layer involves two linearly connected networks with ReLU activations, transforming $X_{\text{average}}$ into ($X_{\text{predict}}$ according to the folowing equation:

$$X_{\text{predict}} = \max(0, X_{\text{average}} W1 + b1) W2 + b2$$

Despite its straightforward nature, this layer proves effective through the GEM, even without a transformer encoder layer. For binary classification, the output $X_{\text{predict}}$ is processed by a sigmoid function $S(X_{\text{predict}}) = \frac{1}{1+e^{-X_{\text{predict}}}}$, and optimization is managed by the Adaptive Momentum Estimation algorithm [14].

## 4.5 Contributions

### 4.5.1 Evaluation of GRN Inference Methods

This work focuses on evaluating GRN inference methods using experimental single-cell RNA-seq datasets, specifically hESC, hHEP, mESC, and mDC. Notably, scGeneRAI has not been tested on these datasets; rather, it has primarily been examined on curated network datasets derived from published models of gene regulatory networks. This evaluation aims to provide a comprehensive comparison of the methods' performance across these experimental conditions. It is anticipated that the supervised approaches will perform significantly better in terms of AUROC and AUPRC metrics, as they leverage training and validation data derived from ground truth. Additionally, it is crucial to consider computational time as a performance parameter, as it reflects the efficiency of the methods in practical applications. Efficient algorithms are more suitable for large-scale datasets commonly encountered in genomic studies.

### 4.5.2 Variability in Ground-Truth GRNs

Ground-truth GRNs, such as STRING, cell-type-specific, and nonspecific networks, encompass different sets of genes. This variability is crucial to consider when developing and applying GRN inference methods. In supervised learning approaches, the training process focuses on genes relevant to the specific networks, as the datasets are curated to include only pertinent genes. This curation ensures that the learning process is focused solely on relevant gene interactions. However, unsupervised methods, which do not rely on predefined labels or curated training sets, lack this inherent filtering mechanism. As a result, when applying unsupervised methods to infer GRNs, there is a risk that the predicted relationships might include genes that are irrelevant or absent in certain specific networks. To address this, it is essential to modify unsupervised methods to incorporate a filtering step that excludes genes that are not relevant to the specific network being studied. After the initial network inference, predicted relationships can be filtered to exclude any gene pairs that involve genes not found in the specific network's

gene set. This step refines the predictions and enhances the relevance of the inferred network. Following the application of GENIE3 and scGeneRAI, the resulting gene pairs will be filtered according to each network type and subsequently reevaluated to ensure the accuracy and relevance of the inferred relationships.

### 4.5.3 Datasets Derived from Expression Data

Traditional literature-based datasets used for training gene regulatory network inference algorithms often suffer from incomplete or biased data, as our knowledge of gene interactions is still evolving. To address these limitations, deriving training and validation datasets directly from expression data offers a more reliable alternative. Unlike literature-based datasets, expression data captures direct observations of gene activity and interactions, making it less susceptible to research biases. The main idea involves using Pearson correlation data between gene pairs to build the training and validation datasets. This ensures that the datasets are directly derived from real biological evidence. The correlations are converted to absolute values to emphasize the strength of the correlation rather than its direction. The strongest (those greater than or equal to the 55th percentile) and weakest (those below or equal to the 45th percentile) correlations are selected, with the strongest labeled as "1" and the weakest as "0." GNNLink was used as the core algorithm while utilizing training and validation datasets derived from Pearson correlation analysis. To enhance the relevance and accuracy of the training and validation data, several refinements were made to the datasets derived from Pearson correlation of gene pairs. These refinements aimed to focus on biologically significant relationships and adapt the datasets for specific network types. The following steps outline the various methods applied and how they were used to train the GNNLink algorithm:

1. **Initial Variation:** The first step involved running GNNLink directly on the training and validation sets derived from the Pearson correlation of gene pairs. This provided a baseline result, reflecting the initial associations inferred from the expression data.

2. **Filtering Based on Transcription Factors:** In the next step, the initial results were refined by filtering out cases where the first gene in the pair was not a transcription factor before labeling. This step was crucial in focusing on regulatory relationships. GNNLink was then executed again on this refined training and validation data.

3. **Network Type-Specific Filtering:** Recognizing that different network types (e.g., Specific, Nonspecific, STRING, LOF/GOF) involve distinct gene sets, a further refinement was applied. For each network type, a list of relevant genes was compiled, and the training and validation data were filtered accordingly to retain only pairs relevant to each

network type. GNNLink was then run again, resulting in network type-specific predictions.

### 4.5.4 Effect of TF Frequency on GRN Inference

Transcription factors exhibit variable prevalence in biological systems; some TFs are highly prevalent and regulate a wide range of genes, while others are less common [58]. Failing to account for this variability can skew predictions, potentially leading to inaccurate representations of gene relationships where the influence of less prevalent TFs is either exaggerated or underestimated. The modification of the GNNLink approach aimed to address this issue by integrating TF frequency data into the prediction process. By incorporating the prevalence of different TFs, the updated method seeks to more accurately reflect their influence on gene relationships. This enhancement ensures that the predictions are better aligned with the actual distribution and regulatory impact of TFs, potentially leading to more precise and reliable predictions of gene interactions.

Baseline models indicated potential to enhance the performance of GRN inference by incorporating additional information about transcription factors. Specifically, leveraging the frequency of transcription factor occurrences across different networks could provide valuable context for improving predictions. To explore this potential, modifications were made to the GNNLink model to incorporate lookup tables containing this information. The first step in this modification involved creating lookup tables that store the raw counts of occurrences for each transcription factor across various networks. These lookup tables serve as an additional input to the model, providing it with prior information that could be beneficial during the learning process. To integrate these lookup tables into the GNNLink model, modifications were made to the data loading process. After loading and normalizing the expression data, an additional step was introduced to incorporate the information from the lookup tables. This involved adjusting the feature matrix used for training the model.

In this step, the feature matrix is adjusted by adding values from the lookup table to each row of the normalized expression data. The function `lookup_dict.get(geneName[i], 0)` retrieves the corresponding value from the lookup table for each gene, identified by `geneName[i]`. If a gene does not exist in the lookup table, a default value of `0` is used. This approach ensures that the model receives additional contextual information about the transcription factors' occurrences as part of its input features.

Building on the initial approach of integrating raw count lookup tables, further experimentation was conducted to refine the model by incorporating TF frequencies rather than raw counts. In this approach, a lookup table of TF frequencies was utilized to adjust gene expression values based on percentiles derived from these frequencies. After normalizing the gene expression data, each gene's expression value was modified according to its associated TF frequency percentile. Specifically, genes with higher TF frequencies received increased expression values, while those with lower frequencies had reduced values. The modification involved the following steps:

1. **Loading and Normalizing Data:** Gene expression data were loaded and normalized as usual.
2. **Frequency Lookup Table:** A lookup table of TF frequencies was read and used to determine the percentile thresholds.
3. **Feature Adjustment:** Expression values were adjusted based on the TF frequency percentiles. Genes with TF frequencies above specific percentiles had their expression values increased by a multiplication factor proportional to the percentile, whereas genes below certain thresholds had their values decreased.

To determine the optimal multiplication factor, a grid search was conducted where each factor was evaluated by training the model and recording the AUC values over several epochs. The performance of each factor was assessed based on the last epoch's AUC. The factor with the highest last epoch AUC was selected as the best, ensuring that the chosen factor optimized the model's performance in leveraging TF frequencies for GRN inference.

# 5 Experimental Results

This chapter begins by introducing the metrics used for the evaluation of the various methods. The evaluation metrics serve as a means to assess the performance of the models in predicting gene regulatory relationships. The results are then presented for the evaluation of the methods in their original form, including GENIE3, scGeneRAI, GNNLink, and STGRNS. Following this, results are provided for the various modifications and further experiments conducted. Each model was run five times on every dataset to evaluate performance consistency and account for potential variability in the results. This repetition enabled the calculation of standard deviation. The standard deviation values are shown below the means in the result tables. Further analysis focused on identifying significant changes in performance, distinguishing meaningful improvements from random fluctuations within one standard deviation.

## 5.1 Chosen Evaluation Metrics

The simplest way to compare a ground truth network with its target is by examining the topology: which nodes are connected to which others? A common method for evaluating a proposed network is to check if the inferred connections between nodes are present or absent in the ground truth network [59]. In the context of GRN inference, the concepts of true positives, false positives, true negatives, and false negatives are crucial for evaluating the accuracy of predicted regulatory interactions between genes. Here's how each term applies:

- **True Positive (TP):** These are correctly predicted regulatory interactions. For example, if the model predicts that Gene A regulates Gene B, and this relationship is confirmed by experimental data (ground truth), it counts as a true positive.
- **True Negative (TN):** These are correctly predicted non-interactions. If the model predicts that Gene A does not regulate Gene B, and this lack of interaction is confirmed by ground truth data, it counts as a true negative.
- **False Positive (FP):** These are incorrect predictions where the model predicts a regulatory interaction between two genes that does not actually exist according to the ground truth data. For instance, if the model predicts that Gene A regulates Gene B, but experimental data shows that there is no such interaction, this is a false positive.
- **False Negative (FN):** These are incorrect predictions where the model fails to predict an existing regulatory interaction. For example, if Gene A is known to regulate Gene B according to experimental data, but the model does not predict this interaction, it counts as a false negative.

This terminology relies on a binary classification of edges, meaning it focuses on whether an edge exists in the network or not. This method is generally sufficient since it works for both directed and undirected networks. However, when distinguishing between activating and inhibiting effects, the same classification can be extended to three categories: 'activation,' 'inhibition,' or 'no effect.' For example, if an activation is predicted but an inhibition is actually expected, it would be considered a false positive [60]. Although this tripartite classification could enhance the analysis by identifying the nature of regulatory interactions, this thesis will refrain from adopting this extension. Instead, the focus will remain on the simpler binary classification. This approach allows for concentrating on the presence or absence of relationships without specifying their regulatory effect.

An essential aspect of assessing a GRN inference algorithm is selecting an appropriate metric. A naive approach might involve setting a threshold on the algorithm's outputs and calculating the average accuracy for identifying edges' presence or absence. However, this method is flawed because GRNs tend to be sparse, allowing an algorithm that consistently predicts the absence of edges to achieve deceptively high accuracy. A more effective approach is to focus on the ratio of true positive predictions to all actual positives (sensitivity or recall) and the ratio of true positive predictions to all predicted positives (precision or positive predictive value) [18].

When inferring a GRN, two common approaches are used for evaluating the reliability of predictions. First, edges can be ranked based on their reliability. Second, the parameters of the learning algorithm can be adjusted to generate networks with varying levels of connectivity.

The performance of the inference method can then be represented through a precision-recall curve (PRC), which is generated by increasing the number of predicted edges according to one of these approaches. Similarly, receiver operating characteristic curve (ROC) can also be used for evaluation. Both the PRC and ROC curves have their own benefits and limitations, which is why they are often used in tandem to assess different inference algorithms. ROC analysis is primarily suitable for binary classification tasks and allows for a direct comparison with random predictions by calculating AUROC. An AUROC close to 0.5 indicates random performance, while values above 0.8 are considered good and below 0.7 are deemed poor. However, because GRNs are often sparse, the number of false positives can significantly exceed the number of true positives. This makes specificity (1 – FPR), commonly used in ROC analysis, less appropriate, as even slight decreases in specificity can lead to a large number of false positives. For this reason, the PRC curve is often a more suitable metric for evaluating GRN inference performance [60].

## 5.2 GRN Inference Methods Assessment

As discussed in Section 4.5.1, evaluation of various established GRN inference methods, including GENIE3, scGeneRAI, GNNLink, and STGRNS, is conducted to establish a baseline for comparative analysis and performance benchmarking. Each method is tested in its original form, and the results, including AUROC and AUPRC metrics, are presented in Table 5.1 and Table 5.3 for TF+500 datasets and in Table 5.2 and Table 5.4 for TF+1000 datasets. It can be observed that the unsupervised methods, GENIE3 and scGeneRAI, perform significantly worse compared to the supervised methods, GNNLink and STGRNS.

The AUROC and AUPRC values for GENIE3 and scGeneRAI across different network types (introduced in Section 3.3) for TF+500 datasets are summarized in Table 5.1. When applied to the TF+500 datasets, GENIE3's AUROC values hover between 0.503 (LOF/GOF, mESC) and 0.671 (STRING, mDC) depending on the specific dataset and network type, with the highest value observed in the STRING network. Similarly, scGeneRAI shows AUROC values ranging from 0.445 (Specific, hHEP) to 0.590 (Specific, hESC) for the same datasets. These results highlight that while there is some variability in performance depending on the dataset, the overall predictive capability of these unsupervised methods remains limited. The AUPRC values further underscore this trend of weak performance, with values ranking from 0.001 to 0.012. These low precision-recall values indicate that both methods struggle to identify true positive gene interactions from the large number of potential false positives.

| Method | Network | hESC | | mESC | | hHEP | | mDC | |
|---|---|---|---|---|---|---|---|---|---|
| | | auroc | auprc | auroc | auprc | auroc | auprc | auroc | auprc |
| GENIE3 | Specific | 0.510 ±0.001 | 0.003 ±0.00005 | 0.537 ±0.001 | 0.012 ±0.00007 | 0.512 ±0.002 | 0.005 ±0.00005 | 0.557 ±0.005 | 0.001 ±0.00001 |
| | Nonspecific | 0.524 ±0.002 | 0.002 ±0.00004 | 0.576 ±0.002 | 0.003 ±0.00004 | 0.508 ±0.002 | 0.002 ±0.00001 | 0.625 ±0.002 | 0.004 ±0.00014 |
| | STRING | 0.652 ±0.002 | 0.005 ±0.00004 | 0.624 ±0.001 | 0.005 ±0.00022 | 0.632 ±0.001 | 0.007 ±0.00010 | 0.671 ±0.002 | 0.008 ±0.00077 |
| | LOF/GOF | - | - | 0.503 ±0.003 | 0.003 ±0.00004 | - | - | - | - |
| scGeneRAI | Specific | 0.590 ±0.015 | 0.004 ±0.00034 | 0.532 ±0.006 | 0.012 ±0.00007 | 0.445 ±0.007 | 0.003 ±0.00007 | 0.520 ±0.014 | 0.001 ±0.00015 |
| | Nonspecific | 0.520 ±0.011 | 0.002 ±0.00012 | 0.495 ±0.007 | 0.002 ±0.00007 | 0.503 ±0.008 | 0.002 ±0.00022 | 0.540 ±0.006 | 0.005 ±0.00026 |
| | STRING | 0.452 ±0.009 | 0.002 ±0.00023 | 0.473 ±0.007 | 0.002 ±0.00022 | 0.459 ±0.006 | 0.003 ±0.00015 | 0.540 ±0.007 | 0.005 ±0.00098 |
| | LOF/GOF | - | - | 0.571 ±0.012 | 0.003 ±0.00045 | - | - | - | - |

**Table 5.1:** AUROC and AUPRC for the unsupervised methods GENIE3 and scGeneRAI on TF+500 datasets show relatively poor performance.

| Method | Network | hESC | | mESC | | hHEP | | mDC | |
|---|---|---|---|---|---|---|---|---|---|
| | | auroc | auprc | auroc | auprc | auroc | auprc | auroc | auprc |
| GENIE3 | Specific | 0.502 ±0.002 | 0.003 ±0.00005 | 0.542 ±0.001 | 0.013 ±0.00005 | 0.514 ±0.004 | 0.005 ±0.00005 | 0.573 ±0.005 | 0.001 ±0.00005 |
| | Nonspecific | 0.510 ±0.001 | 0.002 ±0.00001 | 0.576 ±0.001 | 0.003 ±0.00005 | 0.512 ±0.005 | 0.001 ±0.00004 | 0.598 ±0.001 | 0.003 ±0.00005 |
| | STRING | 0.637 ±0.001 | 0.004 ±0.00001 | 0.646 ±0.001 | 0.005 ±0.00005 | 0.637 ±0.002 | 0.007 ±0.00009 | 0.629 ±0.001 | 0.005 ±0.00005 |
| | LOF/GOF | - | - | 0.504 ±0.001 | 0.003 ±0.00008 | - | - | - | - |
| scGeneRAI | Specific | 0.583 ±0.009 | 0.004 ±0.00013 | 0.540 ±0.011 | 0.008 ±0.00349 | 0.574 ±0.005 | 0.001 ±0.00005 | 0.546 ±0.011 | 0.001 ±0.00005 |
| | Nonspecific | 0.547 ±0.007 | 0.002 ±0.00004 | 0.500 ±0.010 | 0.001 ±0.00040 | 0.483 ±0.016 | 0.001 ±0.00001 | 0.561 ±0.016 | 0.003 ±0.00019 |
| | STRING | 0.451 ±0.006 | 0.002 ±0.00004 | 0.478 ±0.008 | 0.001 ±0.00021 | 0.451 ±0.007 | 0.001 ±0.00004 | 0.526 ±0.012 | 0.004 ±0.00049 |
| | LOF/GOF | - | - | 0.641 ±0.101 | 0.004 ±0.00167 | - | - | - | - |

**Table 5.2:** AUROC and AUPRC results for the unsupervised methods GENIE3 and scGeneRAI on TF+1000 datasets.

The pattern of performance for unsupervised methods GENIE3 and scGeneRAI seen in TF+500 datasets extends to the TF+1000 datasets, as seen in Table 5.2. Although the AUROC and AUPRC values show slight variations, they largely mirror the trends seen in the TF+500 datasets. GENIE3 and scGeneRAI both show a marginal increase in AUROC when applied to larger datasets, but the AUPRC values remain similarly low, reflecting a persistent challenge in identifying accurate regulatory links without leveraging prior biological knowledge.

The performance of GNNLink and STGRNS across different network types (introduced in Section 3.3) can be observed in Table 5.3 for TF+500 datasets and in Table 5.4 for TF+1000 datasets. The analysis is summarized as follows:

1. **Cell-type-specific:** When looking at AUROC values, for the cell-type-specific networks in the mESC and mDC datasets, STGRNS shows superior performance, achieving higher AUROC scores compared to GNNLink in both TF+500 and TF+1000 settings. In contrast, for the cell-type-specific networks in the hHEP dataset, GNNLink outperforms STGRNS. For the hESC dataset, both methods perform similarly, with only minor differences in AUROC values across the TF+500 and TF+1000. When considering AUPRC, STGRNS holds a clear advantage in the cell-type-specific networks of the mESC dataset, displaying significantly higher values across TF+500 and TF+1000 compared to GNNLink. In the hHEP dataset, STGRNS again outperforms GNNLink in AUPRC. However, in the hESC dataset, both methods deliver comparable performance, though STGRNS maintains a slight lead. Notably, for the cell-type-specific networks in the mDC dataset, GNNLink performs better in AUPRC.

2. **Nonspecific:** Across all nonspecific networks, STGRNS consistently achieves better AUROC scores and higher AUPRC values than GNNLink.

3. **STRING:** Across almost all STRING networks, GNNLink demonstrates superior performance in both AUROC and AUPRC compared to STGRNS.

4. **LOF/GOF:** In the context of LOF/GOF networks, GNNLink outperforms STGRNS in terms of both AUROC and AUPRC.

| Method | Network | hESC | | mESC | | hHEP | | mDC | |
|---|---|---|---|---|---|---|---|---|---|
| | | auroc | auprc | auroc | auprc | auroc | auprc | auroc | auprc |
| GNNLink | Specific | 0.807 ±0.007 | 0.448 ±0.020 | 0.837 ±0.008 | 0.712 ±0.004 | 0.889 ±0.006 | 0.593 ±0.006 | 0.576 ±0.059 | 0.728 ±0.004 |
| | Nonspecific | 0.630 ±0.015 | 0.054 ±0.006 | 0.734 ±0.006 | 0.088 ±0.003 | 0.695 ±0.014 | 0.051 ±0.002 | 0.776 ±0.011 | 0.269 ±0.006 |
| | STRING | 0.906 ±0.006 | 0.592 ±0.007 | 0.903 ±0.006 | 0.468 ±0.005 | 0.898 ±0.006 | 0.560 ±0.005 | 0.902 ±0.004 | 0.595 ±0.007 |
| | LOF/GOF | - | - | 0.887 ±0.008 | 0.728 ±0.006 | - | - | - | - |
| STGRNS | Specific | 0.817 ±0.008 | 0.522 ±0.015 | 0.903 ±0.003 | 0.838 ±0.001 | 0.857 ±0.002 | 0.217 ±0.002 | 0.727 ±0.014 | 0.112 ±0.007 |
| | Nonspecific | 0.833 ±0.002 | 0.116 ±0.002 | 0.796 ±0.002 | 0.093 ±0.001 | 0.842 ±0.002 | 0.155 ±0.004 | 0.870 ±0.018 | 0.280 ±0.015 |
| | STRING | 0.808 ±0.007 | 0.226 ±0.026 | 0.766 ±0.004 | 0.112 ±0.001 | 0.862 ±0.005 | 0.328 ±0.012 | 0.889 ±0.009 | 0.564 ±0.022 |
| | LOF/GOF | - | - | 0.792 ±0.005 | 0.467 ±0.008 | - | - | - | - |

**Table 5.3:** AUROC and AUPRC results for the supervised methods STGRNS and GNNLink on TF+500 datasets. STGRNS excels in nonspecific networks, GNNLink outperforms in STRING and LOF/GOF networks, while cell-specific results vary.

| Method | Network | hESC | | mESC | | hHEP | | mDC | |
|---|---|---|---|---|---|---|---|---|---|
| | | auroc | auprc | auroc | auprc | auroc | auprc | auroc | auprc |
| GNNLink | Specific | 0.836 ±0.010 | 0.475 ±0.001 | 0.869 ±0.003 | 0.766 ±0.002 | 0.899 ±0.002 | 0.758 ±0.004 | 0.689 ±0.034 | 0.442 ±0.005 |
| | Nonspecific | 0.658 ±0.008 | 0.064 ±0.004 | 0.736 ±0.007 | 0.090 ±0.006 | 0.691 ±0.004 | 0.035 ±0.005 | 0.772 ±0.014 | 0.276 ±0.006 |
| | STRING | 0.904 ±0.002 | 0.640 ±0.005 | 0.897 ±0.004 | 0.500 ±0.003 | 0.886 ±0.011 | 0.542 ±0.002 | 0.888 ±0.003 | 0.630 ±0.007 |
| | LOF/GOF | - | - | 0.920 ±0.003 | 0.759 ±0.004 | - | - | - | - |
| STGRNS | Specific | 0.834 ±0.005 | 0.514 ±0.012 | 0.910 ±0.001 | 0.851 ±0.002 | 0.861 ±0.002 | 0.797 ±0.004 | 0.777 ±0.014 | 0.238 ±0.013 |
| | Nonspecific | 0.862 ±0.006 | 0.145 ±0.006 | 0.809 ±0.003 | 0.110 ±0.002 | 0.887 ±0.003 | 0.191 ±0.004 | 0.886 ±0.005 | 0.281 ±0.013 |
| | STRING | 0.834 ±0.002 | 0.270 ±0.008 | 0.783 ±0.002 | 0.101 ±0.006 | 0.859 ±0.005 | 0.309 ±0.007 | 0.896 ±0.002 | 0.543 ±0.019 |
| | LOF/GOF | - | - | 0.792 ±0.005 | 0.451 ±0.007 | - | - | - | - |

**Table 5.4:** AUROC and AUPRC for the supervised methods STGRNS and GNNLink on TF+1000 datasets exhibit patterns similar to those in the TF+500 datasets.

In evaluating the performance of the different methods, it is important to note the differences in running times. Running times for each method can be observed in Table 5.5 and Table 5.6. GNNLink demonstrates impressive efficiency, completing tasks involving TFs+500 genes and TFs+1000 genes in a matter of seconds. In contrast, STGRNS can take hours to run, depending on the dataset. This stark difference in execution time makes GNNLink a more practical choice for large-scale analyses. Given its rapid execution and competitive performance across various metrics, GNNLink is well-suited for further improvements in gene regulatory network inference. Its efficiency not only facilitates quicker analyses but also allows for the exploration of larger datasets without the prohibitive time costs associated with other methods. Thus, GNNLink stands out as a robust tool for inferring regulatory relationships, providing a strong foundation for ongoing advancements in this field.

| **Method** | **hESC TF+500** | **mESC TF+500** | **hHEP TF+500** | **mDC TF+500** |
|---|---|---|---|---|
| GENIE3 | 3h 6m 13s | 2h 10m 56s | 1h 19m 43s | 1h 13m 32s |
| scGeneRAI | 14h 0m 20s | 1 d 1h 35m 57s | 15h 29m 25s | 10h 21m 03s |
| STGRNS | 2h 36m 10s | 3h 44m 52s | 2h 26m 27s | 1h 11m 47s |
| GNNLink | 8s | 12s | 10s | 6s |

**Table 5.5:** Running times for different methods (GENIE3, scGeneRAI, STGRNS, and GNNLink) on the TF+500 datasets. For supervised methods (GNNLink and STGRNS), the reported times represent the average across different network types.

| **Method** | **hESC TF+1000** | **mESC TF+1000** | **hHEP TF+1000** | **mDC TF+1000** |
|---|---|---|---|---|
| GENIE3 | 5h 4m 40s | 3h 55m 46s | 2h 27m 28s | 2h 21m 40s |
| scGeneRAI | 2d 2h 20m 38s | 3d 5h 49m 11s | 2d 4h 36m 12s | 1d 15h 51m 26s |
| STGRNS | 4h 10m 14s | 5h 4m 26s | 7h 10m 34s | 2h 15m 27s |
| GNNLink | 12s | 15s | 14s | 8s |

**Table 5.6:** Table showing the running times for different methods (GENIE3, scGeneRAI, STGRNS, and GNNLink) on the TF+1000 datasets. For supervised methods (GNNLink and STGRNS), the reported times represent the average across different network types.

## 5.3 Variability in Ground-Truth GRNs

Following the application of GENIE3 and scGeneRAI, the resulting gene pairs are filtered and reevaluated, as detailed in Section 4.5.2. The results obtained after filtering irrelevant genes from two unsupervised methods, GENIE3 and scGeneRAI, are presented in Tables 5.7 and 5.9 for TF+500 datasets and Tables 5.8 and 5.10 for TF+1000 datasets.

These tables demonstrate the impact of applying a gene pair filtering step, implemented after the GRN inference process, to unsupervised GRN inference methods. Gene pairs were removed if they contained a gene that could not possibly appear in ground truth of a particular network type. The filtering step has a limited impact on the AUROC and AUPRC scores. While there are small adjustments in these metrics, the overall performance of the prediction remains largely unaffected by the filtering process. This is likely because the models themselves are not effectively capturing the task (evidenced by their performance being close to that of a random classifier). In this context, the filtering of irrelevant genes does not substantially improve the results, as the model's predictions are already far from optimal.

An important observation is that, while the changes in results may not be drastic, they provide a more accurate reflection of the model's performance. It is counterintuitive to include genes that are definitively absent from the ground-truth GRNs being tested, as those genes will not contribute to meaningful predictions. Filtering these genes better aligns the evaluation with the true biological networks of interest. However, it's crucial to consider that these genes are part of the single-cell expression datasets used for training. By excluding them, there is a risk of inadvertently introducing a "supervised" element into an otherwise unsupervised method. This occurs because filtering genes based on prior knowledge of the test networks could make the approach more aligned with a supervised learning paradigm.

| Method | Network | hESC | | mESC | | hHEP | | mDC | |
|---|---|---|---|---|---|---|---|---|---|
| | | auroc | auprc | auroc | auprc | auroc | auprc | auroc | auprc |
| GENIE3 | Specific | 0.510 ±0.001 | 0.003 ±0.00005 | 0.537 ±0.001 | 0.012 ±0.00007 | 0.512 ±0.002 | 0.005 ±0.00005 | 0.557 ±0.005 | 0.001 ±0.00001 |
| | Nonspecific | 0.524 ±0.002 | 0.002 ±0.00004 | 0.576 ±0.002 | 0.003 ±0.00004 | 0.508 ±0.002 | 0.002 ±0.00001 | 0.625 ±0.002 | 0.004 ±0.00014 |
| | STRING | 0.652 ±0.002 | 0.005 ±0.00004 | 0.624 ±0.001 | 0.005 ±0.00022 | 0.632 ±0.001 | 0.007 ±0.00010 | 0.671 ±0.002 | 0.008 ±0.00077 |
| | LOF/GOF | - | - | 0.503 ±0.003 | 0.003 ±0.00004 | - | - | - | - |
| GENIE3 filtered | Specific | 0.504 ±0.001 | 0.003 ±0.00001 | 0.527 ±0.001 | 0.014 ±0.00011 | 0.506 ±0.001 | 0.005 ±0.00001 | 0.556 ±0.001 | 0.001 ±0.00010 |
| | Nonspecific | 0.518 ±0.003 | 0.002 ±0.00001 | 0.565 ±0.001 | 0.004 ±0.00004 | 0.508 ±0.001 | 0.002 ±0.00009 | 0.625 ±0.001 | 0.004 ±0.00010 |
| | STRING | 0.648 ±0.001 | 0.007 ±0.00003 | 0.605 ±0.001 | 0.007 ±0.00004 | 0.635 ±0.001 | 0.008 ±0.00007 | 0.679 ±0.001 | 0.010 ±0.00002 |
| | LOF/GOF | - | - | 0.501 ±0.001 | 0.004 ±0.00003 | - | - | - | - |

**Table 5.7:** AUROC and AUPRC for the TF+500 datasets, showcasing the performance of the original GENIE3 and GENIE3 after refining datasets to exclude irrelevant genes.

| Method | Network | hESC | | mESC | | hHEP | | mDC | |
|---|---|---|---|---|---|---|---|---|---|
| | | auroc | auprc | auroc | auprc | auroc | auprc | auroc | auprc |
| GENIE3 | Specific | 0.502 ±0.002 | 0.003 ±0.00005 | 0.542 ±0.001 | 0.013 ±0.00005 | 0.514 ±0.004 | 0.005 ±0.00005 | 0.573 ±0.005 | 0.001 ±0.00005 |
| | Nonspecific | 0.510 ±0.001 | 0.002 ±0.00001 | 0.576 ±0.001 | 0.003 ±0.00005 | 0.512 ±0.005 | 0.001 ±0.00004 | 0.598 ±0.001 | 0.003 ±0.00005 |
| | STRING | 0.637 ±0.001 | 0.004 ±0.00001 | 0.646 ±0.001 | 0.005 ±0.00005 | 0.637 ±0.002 | 0.007 ±0.00009 | 0.629 ±0.001 | 0.005 ±0.00005 |
| | LOF/GOF | - | - | 0.504 ±0.001 | 0.003 ±0.00008 | - | - | - | - |
| GENIE3 filtered | Specific | 0.496 ±0.002 | 0.003 ±0.00004 | 0.534 ±0.002 | 0.016 ±0.00004 | 0.510 ±0.004 | 0.006 ±0.00005 | 0.572 ±0.006 | 0.001 ±0.00009 |
| | Nonspecific | 0.505 ±0.001 | 0.002 ±0.00001 | 0.567 ±0.001 | 0.003 ±0.00004 | 0.509 ±0.005 | 0.002 ±0.00003 | 0.597 ±0.001 | 0.004 ±0.00006 |
| | STRING | 0.631 ±0.001 | 0.006 ±0.00002 | 0.627 ±0.001 | 0.007 ±0.00004 | 0.636 ±0.002 | 0.009 ±0.00008 | 0.638 ±0.002 | 0.006 ±0.00004 |
| | LOF/GOF | - | - | 0.501 ±0.001 | 0.003 ±0.00009 | - | - | - | - |

**Table 5.8:** AUROC and AUPRC for TF+1000 datasets, comparing the performance of the original GENIE3 model with GENIE3 results after excluding irrelevant genes.

| Method | Network | hESC | | mESC | | hHEP | | mDC | |
|---|---|---|---|---|---|---|---|---|---|
| | | auroc | auprc | auroc | auprc | auroc | auprc | auroc | auprc |
| scGeneRAI | Specific | 0.590 ±0.015 | 0.004 ±0.00034 | 0.532 ±0.006 | 0.012 ±0.00007 | 0.445 ±0.007 | 0.003 ±0.00007 | 0.520 ±0.014 | 0.001 ±0.00015 |
| | Nonspecific | 0.520 ±0.011 | 0.002 ±0.00012 | 0.495 ±0.007 | 0.002 ±0.00007 | 0.503 ±0.008 | 0.002 ±0.00022 | 0.540 ±0.006 | 0.005 ±0.00026 |
| | STRING | 0.452 ±0.009 | 0.002 ±0.00023 | 0.473 ±0.007 | 0.002 ±0.00022 | 0.459 ±0.006 | 0.003 ±0.00015 | 0.540 ±0.007 | 0.005 ±0.00098 |
| | LOF/GOF | - | - | 0.571 ±0.012 | 0.003 ±0.00045 | - | - | - | - |
| scGeneRAI filtered | Specific | 0.586 ±0.014 | 0.004 ±0.00022 | 0.511 ±0.004 | 0.014 ±0.00008 | 0.450 ±0.006 | 0.004 ±0.00006 | 0.519 ±0.014 | 0.001 ±0.00016 |
| | Nonspecific | 0.514 ±0.012 | 0.002 ±0.00013 | 0.480 ±0.003 | 0.003 ±0.00008 | 0.504 ±0.006 | 0.002 ±0.00021 | 0.546 ±0.008 | 0.005 ±0.00027 |
| | STRING | 0.467 ±0.007 | 0.004 ±0.00021 | 0.460 ±0.007 | 0.005 ±0.00020 | 0.471 ±0.006 | 0.005 ±0.00010 | 0.542 ±0.006 | 0.008 ±0.00096 |
| | LOF/GOF | - | - | 0.538 ±0.010 | 0.004 ±0.00039 | - | - | - | - |

**Table 5.9:** AUROC and AUPRC results for TF+500 datasets, comparing the performance of the original scGeneRAI model with scGeneRAI results after excluding irrelevant genes.

| Method | Network | hESC | | mESC | | hHEP | | mDC | |
|---|---|---|---|---|---|---|---|---|---|
| | | auroc | auprc | auroc | auprc | auroc | auprc | auroc | auprc |
| scGeneRAI | Specific | 0.583 ±0.009 | 0.004 ±0.00013 | 0.540 ±0.011 | 0.008 ±0.00349 | 0.574 ±0.005 | 0.001 ±0.00005 | 0.546 ±0.011 | 0.001 ±0.00005 |
| | Nonspecific | 0.547 ±0.007 | 0.002 ±0.00004 | 0.500 ±0.010 | 0.001 ±0.00040 | 0.483 ±0.016 | 0.001 ±0.00001 | 0.561 ±0.016 | 0.003 ±0.00019 |
| | STRING | 0.451 ±0.006 | 0.002 ±0.00004 | 0.478 ±0.008 | 0.001 ±0.00021 | 0.451 ±0.007 | 0.001 ±0.00004 | 0.526 ±0.012 | 0.004 ±0.00049 |
| | LOF/GOF | - | - | 0.641 ±0.101 | 0.004 ±0.00167 | - | - | - | - |
| scGeneRAI filtered | Specific | 0.582 ±0.008 | 0.005 ±0.00011 | 0.541 ±0.012 | 0.008 ±0.00331 | 0.680 ±0.005 | −0.001 ±0.00006 | 0.547 ±0.009 | 0.002 ±0.00004 |
| | Nonspecific | 0.545 ±0.008 | 0.001 ±0.00002 | 0.499 ±0.009 | 0.001 ±0.00042 | 0.474 ±0.015 | 0.000 ±0.00002 | 0.567 ±0.015 | 0.003 ±0.00016 |
| | STRING | 0.466 ±0.005 | 0.003 ±0.00005 | 0.494 ±0.007 | 0.002 ±0.00020 | 0.468 ±0.006 | 0.001 ±0.00005 | 0.532 ±0.010 | 0.006 ±0.00048 |
| | LOF/GOF | - | - | 0.625 ±0.097 | 0.006 ±0.00155 | - | - | - | - |

**Table 5.10:** AUROC and AUPRC results for TF+1000 datasets, comparing the performance of the original scGeneRAI model with scGeneRAI results after filtering irrelevant genes.

## 5.4 Training Datasets Derived from Expression Data

The following tables summarize the performance of GNNLink across various training and validation datasets derived from Pearson correlation data, as described in Section 4.5.3. Each refinement step is supposed to progressively enhance dataset quality and relevance by focusing on biologically meaningful gene pairs. The first refinement filtered gene pairs to include only those with transcription factors as the first gene in a gene-gene pair, while the next iteration further adapted the datasets by retaining only genes characteristic of specific network types.

For the TF+500 dataset (results shown in Table 5.11), the Network-specific unsupervised GNNLink model demonstrates the highest overall performance across most datasets and network types. The results are detailed as follows:

- **Cell-Specific Network:** Unsupervised GNNLink tends to outperform the network-specific variant in terms of AUROC across most datasets. Conversely, the Network-specific model demonstrates superior performance in AUPRC, reflecting its effectiveness in predicting true positive rates. Despite these differences, both models perform poorly.
- **Nonspecific Network:** Across most datasets, the Network-specific model achieves the highest performance both in terms of AUROC and AUPRC metrics. The GNNLink unsupervised model with TF as the first gene shows competitive performance, while basic unsupervised GNNLink performs the worst.
- **STRING Network:** The unsupervised GNNLink Network-specific model demonstrates superior performance on the STRING network, achieving the highest AUROC values across all cell types. It records the highest AUPRC for hESC and mESC, while both hHEP and mDC datasets achieve a score of 0.056. Unsupervised GNNLink with TF as the first gene reaches a slightly higher AUPRC of 0.057 for the hHEP and mDC datasets.
- **LOF/GOF Network:** The basic GNNLink unsupervised model achieves the highest performance.

| Method | Network | hESC | | mESC | | hHEP | | mDC | |
|---|---|---|---|---|---|---|---|---|---|
| | | auroc | auprc | auroc | auprc | auroc | auprc | auroc | auprc |
| GNNLink unsupervised | Specific | 0.551 ±0.004 | 0.198 ±0.001 | 0.628 ±0.007 | 0.444 ±0.006 | 0.531 ±0.0010 | 0.341 ±0.001 | 0.584 ±0.011 | 0.097 ±0.009 |
| | Nonspecific | 0.434 ±0.001 | 0.013 ±0.002 | 0.556 ±0.002 | 0.018 ±0.002 | 0.488 ±0.016 | 0.014 ±0.001 | 0.475 ±0.004 | 0.016 ±0.003 |
| | STRING | 0.492 ±0.002 | 0.021 ±0.002 | 0.522 ±0.001 | 0.024 ±0.001 | 0.598 ±0.007 | 0.043 ±0.001 | 0.569 ±0.006 | 0.039 ±0.001 |
| | LOF/GOF | - | - | 0.656 ±0.010 | 0.288 ±0.010 | - | - | - | - |
| GNNLink unsupervised TF as first gene | Specific | 0.563 ±0.009 | 0.190 ±0.002 | 0.604 ±0.005 | 0.422 ±0.001 | 0.512 ±0.005 | 0.342 ±0.002 | 0.521 ±0.006 | 0.091 ±0.001 |
| | Nonspecific | 0.508 ±0.006 | 0.016 ±0.001 | 0.622 ±0.007 | 0.023 ±0.002 | 0.571 ±0.006 | 0.018 ±0.001 | 0.580 ±0.007 | 0.026 ±0.001 |
| | STRING | 0.593 ±0.007 | 0.033 ±0.001 | 0.647 ±0.005 | 0.037 ±0.002 | 0.658 ±0.011 | 0.057 ±0.002 | 0.659 ±0.008 | 0.057 ±0.001 |
| | LOF/GOF | - | - | 0.550 ±0.009 | 0.212 ±0.007 | - | - | - | - |
| GNNLink unsupervised Network-specific | Specific | 0.580 ±0.001 | 0.200 ±0.001 | 0.611 ±0.002 | 0.432 ±0.002 | 0.526 ±0.002 | 0.355 ±0.001 | 0.574 ±0.005 | 0.100 ±0.002 |
| | Nonspecific | 0.516 ±0.002 | 0.016 ±0.001 | 0.624 ±0.006 | 0.024 ±0.001 | 0.575 ±0.005 | 0.019 ±0.001 | 0.580 ±0.005 | 0.026 ±0.002 |
| | STRING | 0.628 ±0.002 | 0.038 ±0.001 | 0.670 ±0.005 | 0.042 ±0.001 | 0.669 ±0.002 | 0.056 ±0.002 | 0.668 ±0.006 | 0.056 ±0.001 |
| | LOF/GOF | - | - | 0.572 ±0.008 | 0.222 ±0.006 | - | - | - | - |

**Table 5.11:** AUROC and AUPRC results for TF+500 datasets, where GNNLink was trained and validated on expression-based datasets that underwent iterative refinement. These refinements included an initial filtering to retain gene pairs where transcription factors are the first gene, followed by a final step to focus on genes specific to each network type.

For the TF+1000 dataset (results shown in Table 5.12), each network type favors a different GNNLink configuration:

- **Cell-Specific Network:** Across all datasets, the GNNLink unsupervised with TF as first gene model consistently outperforms the other models in both AUROC and AUPRC metrics.
- **Nonspecific Network:** In three out of four datasets, the GNNLink unsupervised Network-specific model outperforms the others in terms of AUROC. Additionally, it achieves the highest AUPRC in two datasets.
- **STRING Network:** In the STRING, the basic GNNLink unsupervised model consistently outperforms the other models in both AUROC and AUPRC across most datasets.

- **LOF/GOF Network:** For the LOF/GOF network on the mESC dataset, the GNNLink unsupervised with TF as first gene model demonstrates the highest performance in both AUROC and AUPRC.

| Method | Network | hESC | | mESC | | hHEP | | mDC | |
|---|---|---|---|---|---|---|---|---|---|
| | | auroc | auprc | auroc | auprc | auroc | auprc | auroc | auprc |
| GNNLink unsupervised | Specific | 0.569 ±0.005 | 0.200 ±0.002 | 0.568 ±0.007 | 0.388 ±0.002 | 0.477 ±0.001 | 0.315 ±0.001 | 0.488 ±0.005 | 0.078 ±0.001 |
| | Nonspecific | 0.517 ±0.005 | 0.017 ±0.001 | 0.623 ±0.006 | 0.024 ±0.001 | 0.551 ±0.006 | 0.019 ±0.001 | 0.607 ±0.006 | 0.029 ±0.003 |
| | STRING | 0.641 ±0.006 | 0.041 ±0.002 | 0.660 ±0.006 | 0.044 ±0.004 | 0.660 ±0.005 | 0.053 ±0.004 | 0.713 ±0.007 | 0.079 ±0.003 |
| | LOF/GOF | - | - | 0.576 ±0.004 | 0.216 ±0.011 | - | - | - | - |
| GNNLink unsupervised TF as first gene | Specific | 0.588 ±0.006 | 0.247 ±0.012 | 0.632 ±0.006 | 0.425 ±0.010 | 0.518 ±0.005 | 0.335 ±0.007 | 0.584 ±0.005 | 0.096 ±0.001 |
| | Nonspecific | 0.450 ±0.004 | 0.014 ±0.001 | 0.556 ±0.005 | 0.017 ±0.001 | 0.488 ±0.004 | 0.014 ±0.001 | 0.467 ±0.005 | 0.016 ±0.001 |
| | STRING | 0.487 ±0.005 | 0.022 ±0.001 | 0.508 ±0.007 | 0.023 ±0.001 | 0.607 ±0.006 | 0.044 ±0.001 | 0.523 ±0.004 | 0.035 ±0.002 |
| | LOF/GOF | - | - | 0.641 ±0.006 | 0.262 ±0.011 | - | - | - | - |
| GNNLink unsupervised Network-specific | Specific | 0.575 ±0.005 | 0.205 ±0.009 | 0.563 ±0.005 | 0.386 ±0.009 | 0.497 ±0.004 | 0.326 ±0.010 | 0.534 ±0.005 | 0.087 ±0.001 |
| | Nonspecific | 0.525 ±0.007 | 0.017 ±0.001 | 0.634 ±0.009 | 0.025 ±0.001 | 0.571 ±0.011 | 0.018 ±0.001 | 0.567 ±0.010 | 0.030 ±0.002 |
| | STRING | 0.634 ±0.006 | 0.043 ±0.001 | 0.660 ±0.007 | 0.045 ±0.002 | 0.669 ±0.008 | 0.054 ±0.002 | 0.683 ±0.009 | 0.060 ±0.001 |
| | LOF/GOF | - | - | 0.610 ±0.010 | 0.249 ±0.009 | - | - | - | - |

**Table 5.12:** AUROC and AUPRC results for TF+1000 datasets, where GNNLink was trained and validated on expression-based datasets refined through iterative filtering. These refinements first retained gene pairs with transcription factors as the initial gene, followed by a final step to focus on genes specific to each network type.

The performance of GNNLink trained on derived datasets remains close to that of a random classifier in most cases. GNNLink consistently achieves an AUROC above 0.6 for STRING networks only when trained on the most refined training and validation datasets. Refining gene pairs contributes to some performance gains in specific cases. However, this filtering shifts the approach closer to the original supervised methodology, potentially undermining the goal of avoiding reliance on literature-based ground-truth datasets. Thus, the performance benefits may not fully justify reintroducing these supervised elements. When examining the

results from other unsupervised methods, GENIE3 and scGeneRAI, alongside those from the GNNLink trained on unrefined datasets derived from expression data it can be observed that GNNLink consistently outperforms both GENIE3 and scGeneRAI in terms of AUPRC across all datasets and network types. This highlights the model's strength in identifying relevant relationships in gene networks with greater precision, particularly in contexts where recall of true positive interactions is prioritized.

## 5.5 Impact of TF Frequency on GRN Inference

| Method | Network | hESC | | mESC | | hHEP | | mDC | |
|---|---|---|---|---|---|---|---|---|---|
| | | **auroc** | **auprc** | **auroc** | **auprc** | **auroc** | **auprc** | **auroc** | **auprc** |
| Random Classifier | Specific | 0.500 ±0.003 | 0.012 ±0.00004 | 0.500 ±0.001 | 0.042 ±0.00003 | 0.497 ±0.004 | 0.023 ±0.00001 | 0.500 ±0.002 | 0.003 ±0.00005 |
| | Nonspecific | 0.503 ±0.002 | 0.009 ±0.00001 | 0.500 ±0.004 | 0.010 ±0.00002 | 0.504 ±0.003 | 0.010 ±0.00005 | 0.500 ±0.004 | 0.012 ±0.00003 |
| | STRING | 0.495 ±0.004 | 0.011 ±0.00001 | 0.501 ±0.002 | 0.011 ±0.00003 | 0.500 ±0.004 | 0.018 ±0.00001 | 0.495 ±0.003 | 0.018 ±0.00004 |
| | LOF/GOF | - | - | 0.500 ±0.003 | 0.006 ±0.00001 | - | - | - | - |
| Density-Based Classifier | Specific | 0.504 ±0.003 | 0.012 ±0.00005 | 0.500 ±0.002 | 0.042 ±0.00003 | 0.500 ±0.004 | 0.023 ±0.00002 | 0.508 ±0.004 | 0.003 ±0.00005 |
| | Nonspecific | 0.501 ±0.002 | 0.009 ±0.00001 | 0.500 ±0.003 | 0.010 ±0.00004 | 0.500 ±0.002 | 0.010 ±0.00003 | 0.500 ±0.003 | 0.012 ±0.00004 |
| | STRING | 0.500 ±0.003 | 0.011 ±0.00001 | 0.499 ±0.003 | 0.011 ±0.00004 | 0.502 ±0.004 | 0.018 ±0.00002 | 0.499 ±0.002 | 0.018 ±0.00003 |
| | LOF/GOF | - | - | 0.494 ±0.004 | 0.006 ±0.00001 | - | - | - | - |
| Lookup Table-Based Classifier | Specific | 0.534 ±0.003 | 0.036 ±0.00005 | 0.510 ±0.004 | 0.053 ±0.00002 | 0.524 ±0.003 | 0.049 ±0.00003 | 0.559 ±0.005 | 0.016 ±0.00003 |
| | Nonspecific | 0.509 ±0.003 | 0.011 ±0.00004 | 0.504 ±0.002 | 0.010 ±0.00001 | 0.510 ±0.004 | 0.012 ±0.00003 | 0.507 ±0.002 | 0.013 ±0.00004 |
| | STRING | 0.503 ±0.003 | 0.012 ±0.00002 | 0.501 ±0.003 | 0.011 ±0.00003 | 0.502 ±0.002 | 0.018 ±0.00004 | 0.502 ±0.003 | 0.019 ±0.00002 |
| | LOF/GOF | - | - | 0.532 ±0.003 | 0.022 ±0.00004 | - | - | - | - |

**Table 5.13:** Comparative performance of Random, Density-Based, and Lookup Table-Based Classifiers across various networks for TF+500 datasets.

| Method | Network | hESC | | mESC | | hHEP | | mDC | |
|---|---|---|---|---|---|---|---|---|---|
| | | auroc | auprc | auroc | auprc | auroc | auprc | auroc | auprc |
| Random Classifier | Specific | 0.498 ±0.002 | 0.012 ±0.00003 | 0.499 ±0.004 | 0.042 ±0.00001 | 0.501 ±0.002 | 0.024 ±0.00004 | 0.500 ±0.003 | 0.003 ±0.00002 |
| | Nonspecific | 0.505 ±0.003 | 0.008 ±0.00002 | 0.502 ±0.004 | 0.008 ±0.00009 | 0.498 ±0.005 | 0.008 ±0.00002 | 0.504 ±0.003 | 0.009 ±0.00001 |
| | STRING | 0.500 ±0.001 | 0.009 ±0.00003 | 0.499 ±0.002 | 0.008 ±0.00002 | 0.502 ±0.004 | 0.014 ±0.00001 | 0.506 ±0.003 | 0.014 ±0.00006 |
| | LOF/GOF | - | - | 0.498 ±0.001 | 0.006 ±0.00007 | - | - | - | - |
| Density-Based Classifier | Specific | 0.503 ±0.004 | 0.012 ±0.00001 | 0.500 ±0.003 | 0.042 ±0.00003 | 0.499 ±0.002 | 0.024 ±0.00001 | 0.503 ±0.002 | 0.003 ±0.00004 |
| | Nonspecific | 0.500 ±0.001 | 0.008 ±0.00003 | 0.501 ±0.003 | 0.008 ±0.00002 | 0.500 ±0.002 | 0.008 ±0.00004 | 0.500 ±0.001 | 0.009 ±0.00003 |
| | STRING | 0.499 ±0.003 | 0.009 ±0.00001 | 0.500 ±0.002 | 0.008 ±0.00004 | 0.500 ±0.003 | 0.014 ±0.00002 | 0.500 ±0.003 | 0.014 ±0.00001 |
| | LOF/GOF | - | - | 0.505 ±0.004 | 0.009 ±0.00003 | - | - | - | - |
| Lookup Table-Based Classifier | Specific | 0.537 ±0.004 | 0.039 ±0.00002 | 0.510 ±0.002 | 0.053 ±0.00003 | 0.525 ±0.003 | 0.050 ±0.00002 | 0.556 ±0.003 | 0.014 ±0.00004 |
| | Nonspecific | 0.510 ±0.002 | 0.009 ±0.00003 | 0.505 ±0.003 | 0.008 ±0.00002 | 0.512 ±0.002 | 0.011 ±0.00003 | 0.508 ±0.001 | 0.010 ±0.00003 |
| | STRING | 0.503 ±0.002 | 0.009 ±0.00002 | 0.503 ±0.003 | 0.009 ±0.00002 | 0.500 ±0.003 | 0.014 ±0.00003 | 0.503 ±0.002 | 0.014 ±0.00004 |
| | LOF/GOF | - | - | 0.528 ±0.002 | 0.017 ±0.00003 | - | - | - | - |

**Table 5.14:** Comparative performance of Random, Density-Based, and Lookup Table-Based Classifiers across various networks for TF+1000 datasets.

To evaluate the impact of TF frequency (discussed in Section 4.5.4) on prediction accuracy, three baseline models were tested: a random classifier, a density-based classifier, and a lookup table-based classifier.

Each model's performance is summarized in Table 5.13 and Table 5.14. The random classifier assigned gene relationships without considering TF frequency, while the density-based classifier used known network densities for predictions. The lookup table-based classifier adjusted predictions based on TF prevalence data, resulting in the most substantial improvements in both AUROC and AUPRC metrics, particularly in specific networks. This suggested that incorporating the information about TF prevalence can improve GRN inference, hinting at a potential benefit in integrating this approach into existing methods to further explore its effectiveness in improving regulatory network modeling.

To assess whether incorporating a lookup table with raw TF counts results in improvements over the basic GNNLink model, a comparison of their AUROC and AUPRC scores across dif-

ferent networks and cell types is conducted. The following results pertain to the performance of GNNLink and GNNLink TF raw count for TF+500 (shown in Table 5.15):

- **Specific Network**: In hESC dataset GNNLink shows a marginally higher AUROC (0.807) than TF raw count (0.793), with a similar trend in AUPRC (0.448 vs. 0.430). The standard GNNLink model performs better in both metrics. Similarly, for hHEP cell type both AUROC and AUPRC are slightly higher in GNNLink (AUROC: 0.889; AUPRC: 0.726) compared to TF raw count (AUROC: 0.875; AUPRC: 0.701). In the case of mESC, TF raw count performs marginally better on AUROC (0.848 vs. 0.837) and AUPRC (0.723 vs. 0.712). For mDC, while both methods show lower predictive power, TF raw count shows a slight improvement in AUROC (0.601 vs. 0.576) and AUPRC (0.332 vs. 0.324).
- **Nonspecific Network**: Across nonspecific networks, GNNLink shows a consistent advantage in AUPRC. However, GNNLink TF raw count shows an advantage in AUROC scores for hESC, hHEP and mESC.
- **STRING Network**: Both AUROC and AUPRC metrics are generally higher for GNNLink compared to TF raw count.
- **LOF/GOF Network**: GNNLink outperforms GNNLink TF raw count, although their performance is comparable.

Subsequently, the results for GNNLink and GNNLink TF raw count for TF+1000 are presented in Table 5.16:

- **Specific Network**: For hESC GNNLink TF raw count performs slightly better in AUROC (0.844 vs. 0.836) and AUPRC (0.510 vs. 0.475) compared to GNNLink. When looking at hHEP both methods perform similarly in AUROC (0.899 for both) and AUPRC (0.763 for TF raw count vs. 0.758 for GNNLink), indicating negligible difference. For mESC, GNNLink TF raw count performs marginally better in AUROC (0.873 vs. 0.869), while both methods yield an identical AUPRC (0.766). Lastly, in the case of mDC data GNNLink shows higher AUROC (0.689 vs. 0.673) and AUPRC (0.442 vs. 0.409).
- **Nonspecific Network**: AUPRC values generally show GNNLink's advantage while according to AUROC results GNNLink TF raw count performs better for hESC, hHEP and mESC.
- **STRING Network**: GNNLink scores higher in AUROC and AUPRC across all cell types.

- **LOF/GOF Network**: mESC: Both GNNLink and TF raw count perform similarly with AUROC (0.920 vs. 0.915) and AUPRC (0.759 vs. 0.753)

Overall, while the addition of the lookup table with raw TF counts offers some marginal benefits in specific contexts, the modifications do not result in substantial or consistent improvements over the standard GNNLink model.

| Method | Network | hESC | | mESC | | hHEP | | mDC | |
|---|---|---|---|---|---|---|---|---|---|
| | | auroc | auprc | auroc | auprc | auroc | auprc | auroc | auprc |
| GNNLink | Specific | 0.807 ±0.007 | 0.448 ±0.020 | 0.837 ±0.008 | 0.712 ±0.004 | 0.889 ±0.006 | 0.593 ±0.006 | 0.576 ±0.059 | 0.728 ±0.004 |
| | Nonspecific | 0.630 ±0.015 | 0.054 ±0.006 | 0.734 ±0.006 | 0.088 ±0.003 | 0.695 ±0.014 | 0.051 ±0.002 | 0.776 ±0.011 | 0.269 ±0.006 |
| | STRING | 0.906 ±0.006 | 0.592 ±0.007 | 0.903 ±0.006 | 0.468 ±0.005 | 0.898 ±0.006 | 0.560 ±0.005 | 0.902 ±0.004 | 0.595 ±0.007 |
| | LOF/GOF | - | - | 0.887 ±0.008 | 0.728 ±0.006 | - | - | - | - |
| GNNLink TF raw count | Specific | 0.793 ±0.004 | 0.430 ±0.005 | 0.848 ±0.009 | 0.723 ±0.008 | 0.875 ±0.004 | 0.701 ±0.003 | 0.601 ±0.005 | 0.332 ±0.002 |
| | Nonspecific | 0.681 ±0.003 | 0.046 ±0.005 | 0.742 ±0.004 | 0.043 ±0.006 | 0.703 ±0.009 | 0.036 ±0.002 | 0.760 ±0.004 | 0.182 ±0.003 |
| | STRING | 0.876 ±0.007 | 0.474 ±0.004 | 0.845 ±0.005 | 0.196 ±0.002 | 0.855 ±0.006 | 0.362 ±0.003 | 0.844 ±0.004 | 0.503 ±0.003 |
| | LOF/GOF | - | - | 0.883 ±0.004 | 0.705 ±0.005 | - | - | - | - |
| GNNLink TF frequency | Specific | 0.973 ±0.005 | 0.902 ±0.007 | 0.908 ±0.004 | 0.895 ±0.006 | 0.971 ±0.003 | 0.952 ±0.005 | 0.989 ±0.003 | 0.826 ±0.007 |
| | Nonspecific | 0.831 ±0.003 | 0.111 ±0.005 | 0.765 ±0.007 | 0.186 ±0.002 | 0.838 ±0.004 | 0.112 ±0.004 | 0.816 ±0.006 | 0.124 ±0.003 |
| | STRING | 0.917 ±0.008 | 0.597 ±0.003 | 0.903 ±0.005 | 0.021 ±0.001 | 0.900 ±0.006 | 0.562 ±0.004 | 0.904 ±0.003 | 0.604 ±0.005 |
| | LOF/GOF | - | - | 0.976 ±0.002 | 0.918 ±0.003 | - | - | - | - |

**Table 5.15:** AUROC and AUPRC results for TF+500 datasets across three GNNLink variations: the original model, lookup table based on raw TF counts, and model with a TF frequency-based lookup table with optimized multiplication factors.

| Method | Network | hESC | | mESC | | hHEP | | mDC | |
|---|---|---|---|---|---|---|---|---|---|
| | | auroc | auprc | auroc | auprc | auroc | auprc | auroc | auprc |
| GNNLink | Specific | 0.836 ±0.010 | 0.475 ±0.001 | 0.869 ±0.003 | 0.766 ±0.002 | 0.899 ±0.002 | 0.758 ±0.004 | 0.689 ±0.034 | 0.442 ±0.005 |
| | Nonspecific | 0.658 ±0.008 | 0.064 ±0.004 | 0.736 ±0.007 | 0.090 ±0.006 | 0.691 ±0.004 | 0.035 ±0.005 | 0.772 ±0.014 | 0.276 ±0.006 |
| | STRING | 0.904 ±0.002 | 0.640 ±0.005 | 0.897 ±0.004 | 0.500 ±0.003 | 0.886 ±0.011 | 0.542 ±0.002 | 0.888 ±0.003 | 0.630 ±0.007 |
| | LOF/GOF | - | - | 0.920 ±0.003 | 0.759 ±0.004 | - | - | - | - |
| GNNLink TF raw count | Specific | 0.844 ±0.004 | 0.510 ±0.005 | 0.873 ±0.008 | 0.766 ±0.002 | 0.899 ±0.009 | 0.763 ±0.006 | 0.673 ±0.015 | 0.409 ±0.004 |
| | Nonspecific | 0.679 ±0.003 | 0.053 ±0.001 | 0.745 ±0.006 | 0.070 ±0.003 | 0.696 ±0.004 | 0.035 ±0.002 | 0.765 ±0.019 | 0.244 ±0.007 |
| | STRING | 0.872 ±0.006 | 0.545 ±0.004 | 0.877 ±0.003 | 0.379 ±0.005 | 0.845 ±0.008 | 0.445 ±0.004 | 0.863 ±0.009 | 0.534 ±0.003 |
| | LOF/GOF | - | - | 0.915 ±0.002 | 0.753 ±0.004 | - | - | - | - |
| GNNLink TF frequency | Specific | 0.978 ±0.009 | 0.915 ±0.008 | 0.954 ±0.007 | 0.915 ±0.005 | 0.982 ±0.003 | 0.965 ±0.014 | 0.982 ±0.005 | 0.857 ±0.002 |
| | Nonspecific | 0.868 ±0.005 | 0.369 ±0.003 | 0.749 ±0.008 | 0.242 ±0.006 | 0.827 ±0.007 | 0.396 ±0.004 | 0.837 ±0.014 | 0.142 ±0.003 |
| | STRING | 0.904 ±0.002 | 0.076 ±0.002 | 0.897 ±0.003 | 0.500 ±0.003 | 0.886 ±0.005 | 0.541 ±0.002 | 0.891 ±0.003 | 0.535 ±0.004 |
| | LOF/GOF | - | - | 0.985 ±0.004 | 0.921 ±0.003 | - | - | - | - |

**Table 5.16:** AUROC and AUPRC results for TF+1000 datasets across three GNNLink variations: the baseline model, lookup table with raw TF counts, and TF frequency-based lookup table with optimized multiplication factors.

The variation of GNNLink with lookup table with TF frequency (shown in Table 5.15 and Table 5.16 in "GNNLink TF frequency" row) demonstrates substantial improvements over the GNNLink with lookup table with TF raw count, especially for Cell-specific, Nonspecific, and LOF/GOF network types. In these cases, both AUROC and AUPRC scores increase significantly. Notably, cell-specific networks achieve high AUROC values above 0.9, with the highest at 0.989 for mDC TF+500, while AUPRC scores also reach around 0.9—a notable improvement compared to the original model's AUPRC range of 0.3-0.8. Nonspecific networks exhibit consistent improvements, especially in AUROC, while LOF/GOF networks, although improved, show slightly more moderate gains. For instance, LOF/GOF networks with TF+500 rise from an AUROC of 0.887 and AUPRC of 0.728 to 0.976 and 0.918, respectively, while TF+1000 increases from 0.920 and 0.759 to 0.985 and 0.921. Conversely, the STRING

network sees minimal improvement, as the optimal multiplication factor remains at 1 or 0 in most cases, effectively neutralizing the TF frequency modification. The best multiplication factors were determined through grid search based on validation results, where the most effective factor was selected.

| Network | hESC | mESC | hHEP | mDC |
|---|---|---|---|---|
| Specific | 200 | 100 | 1000 | 900 |
| Nonspecific | 300 | 1000 | 800 | 200 |
| STRING | 1 | 0 | 1 | 1 |
| LOF/GOF | - | 100 | - | - |

**Table 5.17:** The table shows the grid search results for determining optimal multiplication factors for TF+500 datasets. Expression data is adjusted by modifying each value based on the lookup table of TF frequencies, with the modification scaled by the multiplication factor.

| Network | hESC | mESC | hHEP | mDC |
|---|---|---|---|---|
| Specific | 1000 | 200 | 1000 | 900 |
| Nonspecific | 1000 | 900 | 400 | 200 |
| STRING | 0 | 0 | 1 | 100 |
| LOF/GOF | - | 100 | - | - |

**Table 5.18:** The table shows the grid search results for determining optimal multiplication factors for TF+1000 datasets. Expression data is modified by changing each value based on the lookup table of TF frequencies, with the modification scaled by the multiplication factor.

Based on the grid search results, both cell-specific and nonspecific networks benefit from moderate to high multiplication factors, which consistently enhance performance across various datasets. For LOF/GOF networks, a multiplication factor of 100 emerges as optimal in the mESC dataset, which is the only dataset with this ground truth network type available. While this may be dataset-specific, it still highlights the utility of non-zero factors for these networks. As previously noted, STRING networks perform optimally without modification, a trend reflected in their consistent multiplication factors of 0 or 1, reinforcing the suitability of the basic GNNLink variant for these networks. These findings underscore that incorporating TF frequency information, combined with carefully selected multiplication factors, significantly enhances GNNLink's performance across most datasets, particularly in cell-specific, nonspecific, and LOF/GOF network types.

# 6 Conclusion

This thesis focused on evaluating four key methods for gene regulatory network inference and exploring strategies to enhance their performance. The evaluation of four methods for gene regulatory network inference, GENIE3 and scGeneRAI (unsupervised) and STGRNS and GNNLink (supervised), revealed a clear performance distinction. Supervised methods, GNNLink and STGRNS, consistently outperformed the unsupervised approaches, due to their access to training data with known regulatory pairs, which enhanced the inference capabilities of these models. Among the supervised approaches, both GNNLink and STGRNS demonstrated comparable performance in terms of chosen performance metrics, AUROC and AUPRC. However, a key difference lies in their computational efficiency: while STGRNS requires several hours to complete, GNNLink achieves its results within seconds. This significant difference in runtime positions GNNLink as a highly efficient choice for large-scale gene regulatory network inference.

The first contribution of this work emphasizes the importance of filtering gene-gene relationships post-GRN inference, particularly for unsupervised methods. In TRN inference, it is essential to retain only those interactions where a transcription factor regulates a target gene (where the target gene can either be a regular gene or another transcription factor). Furthermore, it was hypothesized that additional filtering of relationships involving genes not present in the specific ground-truth network being tested would improve the evaluation process. Although this filtering step did not significantly improve models' performance, since the core issue lies in the models' inability to capture the relationships effectively, it does provide a more

appropriate and biologically relevant method of evaluation. However, it is important to note that while this filtering improves the evaluation process, it also introduces an element of prior knowledge.

To overcome the challenges posed by incomplete and biased literature-based datasets, this work explored the use of expression data to directly derive training and validation datasets. By leveraging gene-gene association networks inferred from expression data using Pearson correlation, the datasets are grounded in actual biological evidence, providing a more accurate representation of gene interactions. These expression data-derived datasets were then utilized to train and validate GNNLink. However, this approach faced a critical limitation: generating training and validation datasets using Pearson correlation is itself a GRN inference process. Despite ensuring that the first gene in a pair was a transcription factor and excluding genes irrelevant to specific networks, the resulting datasets proved inadequate for training and validation. As a result, the AUROC performance of this modified GNNLink was poor, aligning with other unsupervised methods like GENIE3 and scGeneRAI. However, a notable distinction was observed in AUPRC, where the unsupervised GNNLink consistently outperformed GENIE3 and scGeneRAI across all datasets and network types.

The final and most successful contribution of this work aimed to test the hypothesis that incorporating transcription factor frequency information would improve GRN inference performance. Considering that transcription factors vary in biological systems, without accounting for this variability, the influence of certain transcription factors may be either overstated or understated, leading to inaccurate GRN predictions. Proposed solution involved enhancing GNNLink by integrating a lookup table of transcription factor frequencies to adjust gene expression values. By modifying gene expression values based on transcription factor frequency, the method accounted for the varying prevalence of transcription factors. Genes with higher transcription factor frequencies were amplified, while those with lower frequencies were reduced. This adjustment resulted in significant improvements in predictions for cell-specific and non-specific networks, underscoring the importance of incorporating transcription factor frequency information in GRN inference. This finding suggests that leveraging transcription factor prevalence could be a valuable enhancement for other supervised methods as well.

Through the course of this thesis, it has been demonstrated that graph neural networks in GRN inference show significant potential, positioning them as a promising direction for future research in the field. In the context of gene regulatory network inference, the insights gained from experimental data emerge as a cornerstone for advancing computational methodologies. The experiments conducted in this thesis underscore the critical role of integrating

biologically relevant knowledge, such as transcription factor prevalence. While technological advancements continue to expand the availability of experimental data, the reliance on high quality data and knowledge derived from biological experiments still remains the key driver for further breakthroughs in GRN inference.

# References


[1] D. P. Clark and N. J. Pazdernik, *Molecular biology*. Elsevier, 2012.

[2] G. T. Network, "Single-cell rna data formats," https://training.galaxyproject.org/training-material/topics/single-cell/tutorials/scrna-data-formats/slides-plain.html, 2023, accessed: 2024-11-12.

[3] F. M. Delgado and F. Gómez-Vela, "Computational methods for gene regulatory networks reconstruction and analysis: A review," *Artificial Intelligence in Medicine*, vol. 95, pp. 133–145, April 2019.

[4] Z.-P. Liu, "Towards precise reconstruction of gene regulatory networks by data integration," *Quantitative Biology*, vol. 6, no. 2, pp. 113–128, 2018.

[5] Y. Uzun, "Approaches for benchmarking single-cell gene regulatory network inference methods," *arXiv preprint arXiv:2307.08463*, 2023.

[6] P. Guruprasad, Y. G. Lee, K. H. Kim, and M. Ruella, "The current landscape of single-cell transcriptomics for cancer immunotherapy," *Journal of Experimental Medicine*, vol. 218, no. 1, p. e20201574, 2020.

[7] D. Kim, A. Tran, H. J. Kim, Y. Lin, J. Y. H. Yang, and P. Yang, "Gene regulatory network reconstruction: harnessing the power of single-cell multi-omic data," *NPJ Systems Biology and Applications*, vol. 9, no. 1, p. 51, 2023.

[8] E. Richardson, R. Trevizani, J. A. Greenbaum, H. Carter, M. Nielsen, and B. Peters, "The receiver operating characteristic curve accurately assesses imbalanced datasets," *Patterns*, vol. 5, no. 6, p. 100994, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666389924001090

[9] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, vol. 17, no. 1, pp. 168–192, 2021, open Access. Article publication date: 30 July 2020, Issue publication date: 4 January 2021. [Online]. Available: https://www.emerald.com/insight/2210-8327.htm

[10] A. Pratapa, A. P. Jalihal, J. N. Law, A. Bharadwaj, and T. Murali, "Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data," *Nature methods*, vol. 17, no. 2, pp. 147–154, 2020.

[11] V. A. Huynh-Thu, A. Irrthum, L. Wehenkel, and P. Geurts, "Inferring regulatory networks from expression data using tree-based methods," *PloS one*, vol. 5, no. 9, p. e12776, 2010.

[12] P. Keyl, P. Bischoff, G. Dernbach, M. Bockmayr, R. Fritz, D. Horst, N. Blüthgen, G. Montavon, K.-R. Müller, and F. Klauschen, "Single-cell gene regulatory network prediction by explainable ai," *Nucleic Acids Research*, vol. 51, no. 4, pp. e20–e20, 2023.

[13] G. Mao, Z. Pang, K. Zuo, Q. Wang, X. Pei, X. Chen, and J. Liu, "Predicting gene regulatory links from single-cell rna-seq data using graph neural networks," *Briefings in Bioinformatics*, vol. 24, no. 6, p. bbad414, 2023.

[14] J. Xu, A. Zhang, F. Liu, and X. Zhang, "Stgrns: an interpretable transformer-based method for inferring gene regulatory networks from single-cell transcriptomic data," *Bioinformatics*, vol. 39, no. 4, p. btad165, 2023.

[15] G.-W. Li and X. S. Xie, "Central dogma at the single-molecule level in living cells," *Nature*, vol. 475, no. 7356, pp. 308–315, 2011.

[16] D. Mercatelli, L. Scalambra, L. Triboli, F. Ray, and F. M. Giorgi, "Gene regulatory network inference resources: A practical overview," *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms*, vol. 1863, no. 6, p. 194430, 2020.

[17] D. R. Larson, R. H. Singer, and D. Zenklusen, "A single molecule view of gene expression," *Trends in cell biology*, vol. 19, no. 11, pp. 630–637, 2009.

[18] V. A. Huynh-Thu and G. Sanguinetti, "Gene regulatory network inference: an introductory survey," *Gene regulatory networks: Methods and protocols*, pp. 1–23, 2019.

[19] A. Brazma and J. Vilo, "Gene expression data analysis," *FEBS letters*, vol. 480, no. 1, pp. 17–24, 2000.

[20] N. Geard, "Modelling gene regulatory networks: Systems biology to complex systems," 2004.

[21] S. A. Lambert, A. Jolma, L. F. Campitelli, P. K. Das, Y. Yin, M. Albu, X. Chen, J. Taipale, T. R. Hughes, and M. T. Weirauch, "The human transcription factors," *Cell*, vol. 172, no. 4, pp. 650–665, 2018.

[22] A. G. Papavassiliou, "Transcription factors," *New England Journal of Medicine*, vol. 332, no. 1, pp. 45–47, 1995.

[23] F. Conte, G. Fiscon, V. Licursi, D. Bizzarri, T. D'Antò, L. Farina, and P. Paci, "A paradigm shift in medicine: A comprehensive review of network-based approaches," *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms*, vol. 1863, no. 6, p. 194416, 2020.

[24] Z. Mousavian, K. Kavousi, and A. Masoudi-Nejad, "Information theory in systems biology. part i: Gene regulatory and metabolic networks," in *Seminars in cell & developmental biology*, vol. 51. Elsevier, 2016, pp. 3–13.

[25] Y. Wang, M. Mashock, Z. Tong, X. Mu, H. Chen, X. Zhou, H. Zhang, G. Zhao, B. Liu, and X. Li, "Changing technologies of rna sequencing and their applications in clinical oncology," *Frontiers in oncology*, vol. 10, p. 447, 2020.

[26] G. Chen, B. Ning, and T. Shi, "Single-cell rna-seq technologies and related computational data analysis," *Frontiers in genetics*, vol. 10, p. 317, 2019.

[27] A. Haque, J. Engel, S. A. Teichmann, and T. Lönnberg, "A practical guide to single-cell rna-sequencing for biomedical research and clinical applications," *Genome medicine*, vol. 9, pp. 1–12, 2017.

[28] P. Langfelder and S. Horvath, "Wgcna: an r package for weighted correlation network analysis," *BMC Bioinformatics*, vol. 9, p. 559, 2008.

[29] J. Ruyssinck, V. A. Huynh-Thu, P. Geurts, T. Dhaene, P. Demeester, and Y. Saeys, "Nimefi: gene regulatory network inference using multiple ensemble feature importance algorithms," *PLoS One*, vol. 9, no. 3, p. e92709, 2014.

[30] N. Wani and K. Raza, "Mkl-grni: A parallel multiple kernel learning approach for supervised inference of large-scale gene regulatory networks," *PeerJ Computer Science*, vol. 7, p. e363, 2021.

[31] H. Khojasteh, A. Khanteymoori, and M. H. Olyaee, "Engrnt: Inference of gene regulatory networks using ensemble methods and topological feature extraction," *Informatics in Medicine Unlocked*, vol. 27, p. 100773, 2021.

[32] W.-Y. Loh, "Classification and regression trees," *Wiley interdisciplinary reviews: data mining and knowledge discovery*, vol. 1, no. 1, pp. 14–23, 2011.

[33] L. E. Chai, S. K. Loh, S. T. Low, M. S. Mohamad, S. Deris, and Z. Zakaria, "A review on the computational approaches for gene regulatory network construction," *Computers in biology and medicine*, vol. 48, pp. 55–65, 2014.

[34] J. Wang, A. Ma, Q. Ma, D. Xu, and T. Joshi, "Inductive inference of gene regulatory network using supervised and semi-supervised graph neural networks," *Computational and structural biotechnology journal*, vol. 18, pp. 3335–3343, 2020.

[35] A. Jereesh, G. S. Kumar *et al.*, "Reconstruction of gene regulatory networks using graph neural networks," *Applied Soft Computing*, vol. 163, p. 111899, 2024.

[36] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[37] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Information sciences*, vol. 250, pp. 113–141, 2013.

[38] S. Boughorbel, F. Jarray, and M. El-Anbari, "Optimal classifier for imbalanced data using matthews correlation coefficient metric," *PloS one*, vol. 12, no. 6, p. e0177678, 2017.

[39] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 233–240.

[40] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," *arXiv preprint arXiv:2010.16061*, 2020.

[41] K. Boyd, K. H. Eng, and C. D. Page, "Area under the precision-recall curve: point estimates and confidence intervals," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13*. Springer, 2013, pp. 451–466.

[42] H. R. Sofaer, J. A. Hoeting, and C. S. Jarnevich, "The area under the precision-recall curve as a performance metric for rare binary events," *Methods in Ecology and Evolution*, vol. 10, no. 4, pp. 565–577, 2019.

[43] T. Hayashi, H. Ozaki, Y. Sasagawa, M. Umeda, H. Danno, and I. Nikaido, "Single-cell full-length total rna sequencing uncovers dynamics of recursive splicing and enhancer rnas," *Nature communications*, vol. 9, no. 1, p. 619, 2018.

[44] A. K. Shalek, R. Satija, J. Shuga, J. J. Trombetta, D. Gennert, D. Lu, P. Chen, R. S. Gertner, J. T. Gaublomme, N. Yosef *et al.*, "Single-cell rna-seq reveals dynamic paracrine control of cellular variation," *Nature*, vol. 510, no. 7505, pp. 363–369, 2014.

[45] J. G. Camp, K. Sekine, T. Gerber, H. Loeffler-Wirth, H. Binder, M. Gac, S. Kanton, J. Kageyama, G. Damm, D. Seehofer *et al.*, "Multilineage communication regulates human liver bud development from pluripotency," *Nature*, vol. 546, no. 7659, pp. 533–538, 2017.

[46] L.-F. Chu, N. Leng, J. Zhang, Z. Hou, D. Mamott, D. T. Vereide, J. Choi, C. Kendziorski, R. Stewart, and J. A. Thomson, "Single-cell rna-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm," *Genome biology*, vol. 17, pp. 1–20, 2016.

[47] P. J. Park, "Chip–seq: advantages and challenges of a maturing technology," *Nature reviews genetics*, vol. 10, no. 10, pp. 669–680, 2009.

[48] T. S. Furey, "Chip–seq and beyond: new and improved methodologies to detect and characterize protein–dna interactions," *Nature Reviews Genetics*, vol. 13, no. 12, pp. 840–852, 2012.

[49] M. Marku and V. Pancaldi, "From time-series transcriptomics to gene regulatory networks: A review on inference methods," *PLOS Computational Biology*, vol. 19, no. 8, p. e1011254, 2023.

[50] C. A. Davis, B. C. Hitz, C. A. Sloan, E. T. Chan, J. M. Davidson, I. Gabdank, J. A. Hilton, K. Jain, U. K. Baymuradov, A. K. Narayanan *et al.*, "The encyclopedia of dna elements (encode): data portal update," *Nucleic acids research*, vol. 46, no. D1, pp. D794–D801, 2018.

[51] S. Oki, T. Ohta, G. Shioi, H. Hatanaka, O. Ogasawara, Y. Okuda, H. Kawaji, R. Nakaki, J. Sese, and C. Meno, "Ch ip-atlas: a data-mining suite powered by full integration of public ch ip-seq data," *EMBO reports*, vol. 19, no. 12, p. e46255, 2018.

[52] H. Xu, C. Baroukh, R. Dannenfelser, E. Y. Chen, C. M. Tan, Y. Kou, Y. E. Kim, I. R. Lemischka, and A. Ma'ayan, "Escape: database for integrating high-content published data collected from human and mouse embryonic stem cells," *Database*, vol. 2013, p. bat045, 2013.

[53] L. Garcia-Alonso, C. H. Holland, M. M. Ibrahim, D. Turei, and J. Saez-Rodriguez, "Benchmark and integration of resources for the estimation of human transcription factor activities," *Genome research*, vol. 29, no. 8, pp. 1363–1375, 2019.

[54] Z.-P. Liu, C. Wu, H. Miao, and H. Wu, "Regnetwork: an integrated database of transcriptional and post-transcriptional regulatory networks in human and mouse," *Database*, vol. 2015, p. bav095, 2015.

[55] H. Han, J.-W. Cho, S. Lee, A. Yun, H. Kim, D. Bae, S. Yang, C. Y. Kim, M. Lee, E. Kim *et al.*, "Trrust v2: an expanded reference database of human and mouse transcriptional regulatory interactions," *Nucleic acids research*, vol. 46, no. D1, pp. D380–D386, 2018.

[56] C. v. Mering, M. Huynen, D. Jaeggi, S. Schmidt, P. Bork, and B. Snel, "String: a database of predicted functional associations between proteins," *Nucleic acids research*, vol. 31, no. 1, pp. 258–261, 2003.

[57] D. Marbach, T. Schaffter, D. Floreano, R. J. Prill, and G. Stolovitzky, "The dream4 in-silico network challenge," *Draft, version 0.3*, 2009.

[58] S. Govindarajan and O. Amster-Choder, "Transcription regulation in bacteria," in *Encyclopedia of Microbiology*, fourth edition ed., T. M. Schmidt, Ed. Oxford: Academic Press, 2019, pp. 441–457. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128012383024624

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

[60] M. Hecker, S. Lambeck, S. Toepfer, E. Van Someren, and R. Guthke, "Gene regulatory network inference: data integration in dynamic models—a review," *Biosystems*, vol. 96, no. 1, pp. 86–103, 2009.

# Acknowledgments

I would like to begin by expressing my heartfelt gratitude to the Big Data Management and Analytics Erasmus Mundus consortium for providing me with this invaluable learning experience. Through this opportunity, I was able to deepen my education and refine my skills, while also broadening my cultural perspective. Meeting inspiring individuals, both fellow students and professors, and living in various countries across Europe allowed me to embrace new cultures and ideas, enriching both my academic and personal growth.

I would also like to express my heartfelt thanks to Professor Esteban Zimányi for his dedication to the program and for fostering such a collaborative learning environment. His commitment to our growth and success has made this experience truly enriching.

I would also like to extend my sincere gratitude to my academic supervisor, Professor Nicolò Navarin, and my university tutor, Professor Gabriele Sales, for their invaluable guidance throughout the thesis research process. Their expertise, encouragement, and patience were essential in shaping my work, and I am truly grateful for their support and mentorship.

I am also deeply grateful to my fellow BDMA students, whose support has been invaluable not only academically but also in navigating the many challenges of studying abroad. Their camaraderie, encouragement, and shared experiences made this journey both manageable and memorable.

Lastly, I would like to thank my family and friends back home for always believing in me and providing unwavering support from afar. Their encouragement and faith in me have been a constant source of strength throughout this journey.