scSGL: Signed Graph Learning for Single-Cell Gene Regulatory Network Inference

Abdullah Karaaslanli[1], Satabdi Saha[2], Selin Aviyente[1], and Tapabrata Maiti[2]

[1]*Department of Electrical and Computer Engineering*
Michigan State University, East Lansing, MI, USA

[2]*Department of Statistics and Probability,*
Michigan State University, East Lansing, MI, USA

Abstract

Elucidating the topology of gene regulatory networks (GRN) from large single-cell RNA sequencing (scRNAseq) datasets, while effectively capturing its inherent cell-cycle heterogeneity, is currently one of the most pressing problems in computational systems biology. Recently, graph learning (GL) approaches based on graph signal processing (GSP) have been developed to infer graph topology from signals defined on graphs. However, existing GL methods are not suitable for learning signed graphs, which represent a characteristic feature of GRNs, as they account for both activating and inhibitory relationships between genes. To this end, we propose a novel signed GL approach, scSGL, that incorporates the similarity and dissimilarity between observed gene expression data to construct gene networks. The proposed approach is formulated as a non-convex optimization problem and solved using an efficient ADMM framework. In our experiments on simulated and real single cell datasets, scSGL compares favorably with other single cell gene regulatory network reconstruction algorithms.

Keywords— Graph Signal Processing, Signed Graph Learning, Gene Regulatory Network Inference

1 Introduction

Gene regulatory networks (GRN) represent fundamental molecular regulatory interactions among genes that establish and maintain all required biological functions characterizing a certain physiological state of a cell in an organism [1]. Cell type identity in an organism is determined by how active transcription factors interact with a set of cis-regulatory regions in the genome and controls the activity of genes by either activation or repression of transcription [2]. Usually, the relationship between these active transcription factors and their target genes characterize GRN's. Due to the inherent causality captured by these meaningful biological interactions in GRN's, genome-wide inference of these networks holds great promise in enhancing the understanding of normal cell physiology, and also in characterizing the molecular compositions of complex diseases [3, 4].

GRN's can be mathematically characterized as graphical models where genes represent nodes and the edges quantify the regulatory relationships between them. GRN reconstruction approaches attempt to infer this regulatory network from high-throughput data using statistical and computational approaches. Multiple methods encompassing varying mathematical concepts have been proposed during the last decade to infer GRN's using gene expression data from bulk population sequencing technologies, which accumulate expression profile from all cells in a tissue. These methods can be broadly classified into two main groups: the first group infers a static GRN, considering steady state of gene expression, while the second group uses temporal measurements to capture the expression profile of the genes

in a dynamic process. A thorough evaluation of the static and dynamic models used in bulk GRN reconstruction can be found in [5, 6].

Recent advances in RNA-sequencing technologies has enabled the measurement of gene expression in single cells. This has led to the development of several computational approaches aimed at quantifying the expression of individual cells for cell-type labelling and estimation of cellular lineages. Several algorithms have been developed to arrange cells in a projected temporal order (pseudotime trajectory) based on similarities in their transcriptional states. In parallel, several dynamic models for single cell GRN reconstruction have also been developed taking into account the estimated pseudotimes. Since single cell network reconstruction algorithms try to establish functional relationships between genes taking into account the entire population of cells, it is debatable as to whether additional knowledge regarding cell state transitions may provide any added benefits [7, 8]. In summary, direct application of bulk GRN reconstruction methods may not be adequate for single cell network inference.

The complex nature of single cell gene expression experiments brings about new computational challenges. Generally, a high percentage of genes are not expressed in single cells, giving rise to the dilemma of whether this "dropout" is due to technical noise or genuine biological variability. Several statistical methods have been designed to model this "dropout" phenomenon [9, 10, 11]. However, recent research has demonstrated that zero inflation models become unnecessary in high-throughput droplet-based scRNA-seq methods where the level of biological zeros is controlled. Additional zero values referred to as dropouts most likely result from biological variation and may be indicative of heterogeneity in gene expression for varying cell types [12, 13]. Based on these insights, we develop a network reconstruction algorithm that learns the co-expression between genes by borrowing ideas from graph signal processing literature. Characterization of these co-expression graphs may help in gaining a better understanding of the causal dependencies present in GRNs.

Graph signal processing (GSP) aims to develop techniques for analysis of signals defined on graphs by extending classical signal processing tools and concepts [14]. In many applications of GSP, the graph topology is not always available, thus it must be inferred from the observed data. Recent work in GSP focuses on learning the graph structure that captures the characteristics of graph signals [15]. The major approaches to graph learning include statistical methods such as Gaussian graphical models; diffusion process based models, where the graph signals are assumed to be stationary random processes and the observed signals are graph filtered versions of these processes; and learning graphs from observations of smooth signals [16]. In this work, we focus on learning graphs with the assumption that graph signals must vary smoothly with respect to graph structure. There are several reasons that motivate this focus. First, smooth signals admit low-pass, band-limited (i.e., sparse) representations using the Graph Fourier basis. Thus, the graph learning problem can be equivalently viewed as one of finding efficient information processing transforms for graph signals. Second, smoothness is a fundamental principle for several graph-based statistical learning tasks including nearest neighbor prediction, denoising, semi-supervised learning, and spectral clustering. Finally, many real-world graph signals are smooth as graphs are constructed based on similarities between nodal attributes [16].

Considering the advantages of GL approaches in learning graph topologies that are consistent with the observed signals, in this paper, we propose a novel GL algorithm for the reconstruction of GRNs. In particular, we assume gene expression data obtained from cells are graph signals residing on an unknown graph structure, which corresponds to the GRN. One important characteristic of GRNs is that they are signed graphs, where positive and negative edges correspond to activating and inhibitory regulations between genes. The effects in the overall system caused by perturbations to regulators can be determined using these signs. Therefore, existing GL approaches, which learn only unsigned graphs, are not suitable for GRN inference. To this end, a novel and computationally efficient signed GL approach, scSGL, that incorporates similarity and dissimilarity within gene expression data is proposed. Signed graphs from gene expression data are either constructed using correlation measures or by estimating the gene covariance matrix after placing strong parametric assumptions on the gene nodes. In an alternative approach, scSGL reconstructs the GRN under the assumption that graph signals admit low-frequency representation over positive

edges, while admitting high-frequency representation over negative edges.

Our main contributions can be summarized as follows: (1) A novel signed GRN inference algorithm, scSGL, is proposed by extending GL approaches to signed graphs. (2) To the best of our knowlege, scSGL is the first GSP based algorithm for single cell GRN reconstruction. (3) scSGL consistently outperforms nine state of the art single cell GRN algorithms (Refer to Section 5) on both simulated and real-world benchmark gene expression datasets.

The remainder of the paper is organized as follows. First, we review related work on GRN inference and existing GL approaches. In Section 3, background on graphs and unsigned graph learning is given. Section 4 describes the proposed signed graph learning problem and the corresponding optimization algorithm. Finally, the performance of scSGL is evaluated and compared to existing GRN inference methods on simulated and experimental gene expression data.

2 Related Works

2.1 Single Cell Gene Regulatory Network Inference

Single cell GRN reconstruction methods, similar to bulk GRN methods, can be broadly classified into static and dynamic models. Both of these groups have extended mathematical concepts explored in bulk GRN reconstruction, to fit the complex and heterogeneous nature of single cell gene expression experiments. Boolean models like BTR [17] and SCNS [18] construct boolean networks and boolean update functions to model the GRN. Differential equation based methods model the gene expression dynamics of a gene as a function of the expression of other genes and environmental factors [19, 20]. SCENIC [21] combines cis-regulatory motif analysis with the random forest based regression analysis framework of GENIE3 [22] to predict the expression profile of each target gene from profiles of all the other genes. PIDC [23] leverages multivariate information theory to estimate the statistical relationships between gene triplets. Pseudotime based correlation ensemble methods like LEAP [24], SCINGE [25] and SINCERITIES [26] calculate the gene correlations over time lag windows and then compute aggregated ranked predictions based on some ensemble machinery. GRNVBEM [27] uses a Bayesian first-order autoregressive moving-average model within a variational Bayes framework for modelling and inferring GRNs. Despite the presence of a multitude of single cell GRN reconstruction methods, recent reviews evaluating the performance of these algorithms on simulated and real experimental datasets, have shown these methods to be less than ideal in recovering the true GRN relationships [28, 8].

2.2 Graph Learning

Dong et al. [29] describe the graph signals using a factor analysis model, where the observed graph signals are assumed to be controlled by a set of unobserved latent variables with some given probabilistic prior. The transformation from the latent variables to the observed signals involves information about the topology of the graph, i.e. the observed signals are low-frequency signals on the underlying network. This model results in an optimization framework with terms that minimize the error between the observed signal and the underlying signal while ensuring that the graph signals are smooth on the underlying network. Kalofolias et al. [30] extended this framework by establishing the link between smoothness and sparsity and adding a regularization term on the degree vector to ensure that each vertex has at least one incident edge. Different variations of these frameworks to handle missing values and sparse outliers in the graph signals were considered in [31, 32, 33, 34, 35]. All of the previous works learn unsigned graphs with the exception of [36], where a signed graph is learned by employing signed graph Laplacian defined in [37]. By using signed Laplacian, [36] aims to learn positive edges between nodes whose signal values are similar and negative edges between nodes whose signal values have opposite signs with similar absolute values. However, this approach is not

suitable when graph signals are either all positive- or negative-valued, as in the case of gene expression data. Our approach overcomes this problem by defining dissimilarities in terms of the distance between graph signals.

3 Background

3.1 Notations

Graphs are represented as $G = (V, E)$ where V is the node set with $|V| = n$ and $E \subseteq V \times V$ is the edge set. An edge between nodes i and j is shown by e_{ij}. In this work, we are interested in *undirected* graphs, that is $e_{ij} = e_{ji}$. Each edge is associated with a weight w_{ij}. The graph is called *unsigned*, if all its weights are non-negative, and *signed*, otherwise. One can represent a graph with an adjacency matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ with entries $A_{ij} = w_{ij}$. If the graph is signed, the adjacency matrix can be written as the summation of two matrices corresponding to the positive and negative edge weights, i.e. $\mathbf{A} = \mathbf{A}^+ - \mathbf{A}^-$ where $A^+_{ij} = w_{ij}$ for $w_{ij} > 0$ and $A^-_{ij} = |w_{ij}|$ for $w_{ij} < 0$. The graph Laplacian is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D}$ is the diagonal degree matrix, i.e. $D_{ii} = \sum_{j=1}^{n} A_{ij}$. For a signed graph, we define two Laplacian matrices $\mathbf{L}^+$ and $\mathbf{L}^-$ which are constructed from $\mathbf{A}^+$ and $\mathbf{A}^-$, respectively. Finally, a *graph signal* $\mathbf{x} \in \mathbb{R}^n$ is defined as a vector whose ith entry is defined the signal value on the ith node of graph G. For scRNAseq, $\mathbf{x}$ corresponds to the expression data of n genes in a cell.

All-one and all-zero vectors and matrices are represented by $\mathbf{1}$ and $\mathbf{0}$, respectively. upper$(\cdot)$ operator takes an $n \times n$ matrix and returns a $n(n-1)/2$-dimensional vector that corresponds to the upper triangular part of the input matrix. The matrix $\mathbf{P} \in \mathbb{R}^{n\times n(n-1)/2}$ is defined such that $\mathbf{P}\text{upper}(\mathbf{A}) = \mathbf{A1}$ where $\mathbf{A}$ is a symmetric matrix whose diagonal entries are equal to zero. Finally, ith row and column of a matrix $\mathbf{X}$ are represented by $\mathbf{X}_{i\cdot}$ and $\mathbf{X}_{\cdot i}$, respectively.

3.2 Unsigned Graph Learning

A graph signal defined on an unsigned graph is said to be smooth if the signal values on connected nodes are similar to each other. There are various measures proposed in GSP literature to quantify the smoothness of a graph signal $\mathbf{x}$ [28, 32]. One common measure is the quadratic form of the graph Laplacian:

$$\mathbf{x}^\top \mathbf{L}\mathbf{x} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}(x_i - x_j)^2. \tag{1}$$

This metric measures the smoothness of the signal with respect to the graph structure. Smaller values of this metric indicate smoother graph signals. Let $\mathbf{X} \in \mathbb{R}^{n\times m}$ be a data matrix whose columns are the graph signals defined on G and m is the number of graph signals. One can learn the Laplacian of G by solving following optimization problem [29]:

$$\min_{\mathbf{L}\in\mathbb{L}} \quad \text{tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}) + \alpha\|\mathbf{L}\|_F^2 \quad \text{s.t.} \quad \text{tr}(\mathbf{L}) = 2n \tag{2}$$

where $\mathbb{L} = \{\mathbf{L} | L_{ij} = L_{ji} \leq 0 \ \forall i \neq j, \mathbf{L1} = \mathbf{0}\}$ is the set of Laplacian matrices, $\|\mathbf{L}\|_F$ is the Frobenius norm and the constraint ensures that trivial solutions are avoided. In this formulation, the first term quantifies the smoothness of the signals with respect to the graph structure and the second term controls the sparsity of the learned graph such that larger α values result in denser graphs.

As noted in [30], (2) can be rewritten in terms of the adjacency matrix $\mathbf{A}$ using the fact that $2\text{tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}) = \text{tr}(\mathbf{A}\mathbf{Z})$

where $\mathbf{Z}$ is the Euclidean distance matrix with its entries $Z_{ij} = \|\mathbf{X}_{i\cdot} - \mathbf{X}_{j\cdot}\|_2^2$:

$$\min_{\mathbf{A}\in\mathbb{A}} \quad \frac{1}{2}\text{tr}(\mathbf{A}\mathbf{Z}) + \alpha\|\mathbf{A}\mathbf{1}\|_2^2 + \alpha\|\mathbf{A}\|_F^2 \text{ s.t. } \mathbf{1}^\top\mathbf{A}\mathbf{1} = 2n, \tag{3}$$

where $\mathbb{A} = \{\mathbf{A}|A_{ij} = A_{ji} \geq 0\ \forall i \neq j, \text{diag}(\mathbf{A}) = \mathbf{0}\}$ is the set of adjacency matrices. This modification also allows the usage of other distance metrics when constructing $\mathbf{Z}$, *e.g.* Manhattan distance as done in [32].

4 Signed Graph Learning

In Section 3.2, an unsigned graph is learned from graph signals by assuming the data is smooth over the graph. In order to learn a signed graph G, one needs to make some additional assumptions about the graph signals $\mathbf{X}$ defined on G. In this work, we make the following assumptions:

- Signal values defined on nodes connected by positive edge values are similar to each other, i.e. Z_{ij} is small for $w_{ij} > 0$.
- Signal values defined on nodes connected by negative edge values are dissimilar to each other, i.e. Z_{ij} is large for $w_{ij} < 0$.

To quantify the similarity and dissimilarity of graph signals over positive and negative edges, we propose to use the Laplacian matrices $\mathbf{L}^+$ and $\mathbf{L}^-$, respectively. In particular, we quantify the similarity over positive edges as $\text{tr}(\mathbf{X}^\top\mathbf{L}^+\mathbf{X}) = 0.5\text{tr}(\mathbf{Z}\mathbf{A}^+)$, whose value decreases as the similarity of graph signals over the positive edges of the graph increases. Similarly, $\mathbf{L}^-$ is employed to quantify the dissimilarity of the graph signal over negative edges as $\text{tr}(\mathbf{X}^\top\mathbf{L}^-\mathbf{X}) = 0.5\text{tr}(\mathbf{Z}\mathbf{A}^-)$, whose value increases as the dissimilarity increases. This idea of using $\mathbf{L}^+$ and $\mathbf{L}^-$ to measure similarities and dissimilarities of graph signals is analogous to a commonly used assumption in community detection for signed graphs. In community detection for signed graphs, it is assumed that nodes within the same community are connected with positive edges, while negative edges fall between communities [38]. The quality of the detected community structure is then quantified by measures similar to $\text{tr}(\mathbf{X}^\top\mathbf{L}^+\mathbf{X})$ and $\text{tr}(\mathbf{X}^\top\mathbf{L}^-\mathbf{X})$.

In order to learn an unknown signed graph from data matrix $\mathbf{X}$, we propose to minimize $\text{tr}(\mathbf{Z}\mathbf{A}^+)$ and maximize $\text{tr}(\mathbf{Z}\mathbf{A}^-)$ with respect to $\mathbf{A}^+$ and $\mathbf{A}^-$, respectively. To this end, we extend the problem in (3) for signed graph learning as follows:

$$\begin{aligned}
\min_{\mathbf{A}^+,\mathbf{A}^-\in\mathbb{A}} \quad & \frac{1}{2}\text{tr}(\mathbf{Z}\mathbf{A}^+) - \frac{1}{2}\text{tr}(\mathbf{Z}\mathbf{A}^-) + \alpha_1\|\mathbf{A}^+\mathbf{1}\|_2^2 + \alpha_1\|\mathbf{A}^+\|_F^2 + \alpha_2\|\mathbf{A}^-\mathbf{1}\|_2^2 + \alpha_2\|\mathbf{A}^-\|_F^2 \\
\text{s.t.} \quad & \mathbf{1}^\top\mathbf{A}^+\mathbf{1} = 2n,\ \mathbf{1}^\top\mathbf{A}^-\mathbf{1} = 2n \\
& A_{ij}^+ = 0 \text{ if } A_{ij}^- \neq 0 \text{ and } A_{ij}^- = 0 \text{ if } A_{ij}^+ \neq 0,
\end{aligned} \tag{4}$$

where the last constraint is imposed to make sure that $\mathbf{A}^+$ and $\mathbf{A}^-$ are not non-zero for the same indices, i.e. an edge cannot be positive and negative, simultaneously. Hyperparameters α_1 and α_2 can separately be tuned to control the sparsity of positive and negative parts of the learned signed graph. To solve this problem, we first rewrite it in vector form. Let $\mathbf{z} = \text{upper}(\mathbf{Z})$, $\mathbf{a}^+ = \text{upper}(\mathbf{A}^+)$ and $\mathbf{a}^- = \text{upper}(\mathbf{A}^-)$, then the problem in (4) can be vectorized as follows:

$$\begin{aligned}
\min_{\mathbf{a}^+,\mathbf{a}^-\geq 0} \quad & \mathbf{z}^\top\mathbf{a}^+ - \mathbf{z}^\top\mathbf{a}^- + \alpha_1\|\mathbf{P}\mathbf{a}^+\|_2^2 + 2\alpha_1\|\mathbf{a}^+\|_2^2 + \alpha_2\|\mathbf{P}\mathbf{a}^-\|_2^2 + 2\alpha_2\|\mathbf{a}^-\|_2^2 \\
\text{s.t.} \quad & \mathbf{1}^\top\mathbf{a}^+ = n,\ \mathbf{1}^\top\mathbf{a}^- = n \\
& \mathbf{a}_i^+ = 0 \text{ if } \mathbf{a}_i^- \neq 0,\ \mathbf{a}_i^- = 0 \text{ if } \mathbf{a}_i^+ \neq 0.
\end{aligned} \tag{5}$$

Since $\mathbf{a}^+ \geq 0$ and $\mathbf{a}^- \geq 0$, the last two constraints can be combined into a single constraint ${\mathbf{a}^+}^\top\mathbf{a}^- = 0$. This type of constraints are called complementarity constraints and make the problem non-convex [39]. Following [40], where

it is shown that alternating direction method of multipliers (ADMM) converges for problems with complementarity constraints under some assumptions, a solution for the above problem using ADMM is provided in the next section.

4.1 Optimization

To able to solve the optimization problem using ADMM, we rewrite the problem in (5) in the standard ADMM form as described in [40]. Introducing the slack variables $\mathbf{v} = \mathbf{a}^+$ and $\mathbf{w} = \mathbf{a}^-$, the standard ADMM form of (5) is:

$$\begin{aligned} \min_{\mathbf{v},\mathbf{w},\mathbf{a}^+,\mathbf{a}^-} \quad & \imath_S(\mathbf{v},\mathbf{w}) + h(\mathbf{a}^+,\mathbf{a}^-) + \imath_H(\mathbf{a}^+) + \imath_H(\mathbf{a}^-) \\ \text{s.t.} \quad & \mathbf{v} - \mathbf{a}^+ = \mathbf{0},\ \mathbf{w} - \mathbf{a}^- = \mathbf{0}, \end{aligned} \tag{6}$$

where $\imath_S()$ is the indicator function for the complementarity set $S = \{(\mathbf{v},\mathbf{w}) : \mathbf{v} \geq 0,\ \mathbf{w} \geq 0,\ \mathbf{v}^\top \mathbf{w} = 0\}$, $h(\mathbf{a}^+,\mathbf{a}^+)$ is the objective function in (5), and $\imath_H()$ is the indicator function for the hyperplane $H = \{\mathbf{a} : \mathbf{1}^\top \mathbf{a} = n\}$. The augmented Lagrangian for this problem is as follows:

$$\begin{aligned} \mathscr{L}_\rho(\mathbf{v},\mathbf{w},\mathbf{a}^+,\mathbf{a}^-,\lambda_1,\lambda_2) = \ & \imath_S(\mathbf{v},\mathbf{w}) + h(\mathbf{a}^+,\mathbf{a}^-) + \imath_H(\mathbf{a}^+) + \imath_H(\mathbf{a}^-) + \lambda_1^\top(\mathbf{v} - \mathbf{a}^+) \\ + \ & \frac{\rho}{2}\|\mathbf{v} - \mathbf{a}^+\|_2^2 + \lambda_2^\top(\mathbf{w} - \mathbf{a}^-) + \frac{\rho}{2}\|\mathbf{w} - \mathbf{a}^-\|_2^2, \end{aligned} \tag{7}$$

where λ_1 and λ_2 are Lagrange multipliers and $\rho > 0$ is the Augmented Lagrangian parameter.

$(\mathbf{v},\mathbf{w})$-step: The $(\mathbf{v},\mathbf{w})$-step of ADMM can be found as the projection onto the complementarity set S:

$$\begin{aligned} (\mathbf{v}^{k+1},\mathbf{w}^{k+1}) &= \operatorname*{argmin}_{\mathbf{v},\mathbf{w}} \imath_S(\mathbf{v},\mathbf{w}) + \frac{\rho}{2}\|\mathbf{v} - \mathbf{a}^{+k} + \frac{\lambda_1^k}{\rho}\|_2^2 \frac{\rho}{2}\|\mathbf{w} - \mathbf{a}^{-k} + \frac{\lambda_2^k}{\rho}\|_2^2 \\ &= \Pi_S(\mathbf{y}), \end{aligned} \tag{8}$$

where $\mathbf{y} = [(\mathbf{a}^{+k} - \lambda_1^k/\rho)^\top, (\mathbf{a}^{-k} - \lambda_2^k/\rho)^\top]^\top$ and $\Pi_S(\cdot)$ is the projection operator on the set S. Projection of $\mathbf{y}$ onto the complementarity set S is as follows. Let $M = n(n-1)/2$,

$$\begin{aligned} v_i = \max(0, y_i) \text{ and } w_i = 0, \text{ if } y_i \geq y_{i+M}, \\ w_i = \max(0, y_{i+M}) \text{ and } v_i = 0, \text{ if } y_i < y_{i+M}. \end{aligned}$$

$(\mathbf{a}^+,\mathbf{a}^-)$-step: Using the fact that optimization can be performed separately for $\mathbf{a}^+$ and $\mathbf{a}^-$, $\mathbf{a}^+$-step can be written as:

$$\begin{aligned} \mathbf{a}^{+k+1} &= \operatorname*{argmin}_{\mathbf{a}^+} \ \mathbf{z}^\top \mathbf{a}^+ + 2\alpha_1\|\mathbf{a}^+\|_2^2 + \alpha_1\|\mathbf{P}\mathbf{a}^+\|_2^2 + \imath_H(\mathbf{a}^+) + \frac{\rho}{2}\|\mathbf{v}^{k+1} - \mathbf{a}^+ + \frac{\lambda_1^k}{\rho}\|_2^2 \\ &= \Pi_H[((4\alpha_1 + \rho)\mathbf{I} + 2\alpha_1\mathbf{P}^\top\mathbf{P})^{-1}(\rho\mathbf{v}^{k+1} + \lambda_1^k - \mathbf{z})], \end{aligned} \tag{9}$$

where $\Pi_H(\cdot)$ is the projection operator on the hyperplane H. Similarly, $\mathbf{a}^-$-step can be found:

$$\mathbf{a}^{-k+1} = \Pi_H[((4\alpha_2 + \rho)\mathbf{I} + 2\alpha_2\mathbf{P}^\top\mathbf{P})^{-1}(\rho\mathbf{w}^{k+1} + \lambda_2^k + \mathbf{z})]. \tag{10}$$

Lagrange multipliers udpate: The updates of Lagrange multipliers are as follows:

$$\lambda_1^{k+1} = \lambda_1^k + \rho(\mathbf{v}^{k+1} - \mathbf{a}^{+k+1}), \tag{11}$$

$$\lambda_2^{k+1} = \lambda_2^k + \rho(\mathbf{w}^{k+1} - \mathbf{a}^{-k+1}). \tag{12}$$

4.2 Convergence

Convergence of this approach can be shown by following the proof in [40], as the problem in (6) satisfies the assumptions stated in [40]. In particular,

Theorem 1 *For a large enough ρ, the sequence $(\mathbf{v}^k, \mathbf{w}^k, \mathbf{a}^{+^k}, \mathbf{a}^{-^k}, \lambda_1^k, \lambda_2^k)$ generated by ADMM converges to a stationary point of the Augmented Lagrangian in* (7).

Proof: The statement can be proven by following [40]. The sketch of the proof is as follows. By Lemma 6 of [40], it can be shown that Augmented Lagrangian is monotonically non-increasing and lower-bounded, which implies $\|\mathbf{v}^k - \mathbf{v}^{k+1}\| + \|\mathbf{w}^k - \mathbf{w}^{k+1}\|$ and $\|\mathbf{a}^{+^k} - \mathbf{a}^{+^{k+1}}\| + \|\mathbf{a}^{-^k} - \mathbf{a}^{-^{k+1}}\|$ go to 0 as $k \to \infty$. By the coercivity of the objective function in (6), the sequence $(\mathbf{v}^k, \mathbf{w}^k, \mathbf{a}^{+^k}, \mathbf{a}^{-^k}, \lambda_1^k, \lambda_2^k)$ is bounded. This implies that there exist a convergent subsequence of $(\mathbf{v}^k, \mathbf{w}^k, \mathbf{a}^{+^k}, \mathbf{a}^{-^k}, \lambda_1^k, \lambda_2^k)$ with limit point $(\mathbf{v}^*, \mathbf{w}^*, \mathbf{a}^{+^*}, \mathbf{a}^{-^*}, \lambda_1^*, \lambda_2^*)$, whose stationarity can be proven by employing Lemma 10 and proof of Theorem 1 given in [40].

5 Results

In this section, the proposed method, scSGL, is compared to eight state-of-the-art GRN inference methods on simulated and experimental scRNAseq datasets. These methods are GENIE3 [22], GRNBoost [41], GRNVBEM [27], GRISLI, SCODE [19], PIDC [23], LEAP [24] and PPCOR [42]. Among these GENIE3, GRNBoost and PPCOR were originally developed for bulk analysis. These methods encompass most of the popular approaches that have been proposed to reconstruct GRN's from high throughput sequencing data (see Section 2.1). Most of the single cell algorithms [19, 20, 25, 27, 24] use cell pseudotime ordering information for algorithm construction with PIDC being an exception. Only three methods [42, 27, 19] among the eight, return signed edges and two methods [42, 23] return undirected edges.

Performance Evaluation: Reference networks are available for both simulated and experimental data. They are used to evaluate the performance of the algorithms. In reality, interactions in GRNs are directed, but since the proposed method and some of the considered methods return undirected graphs, we discard the directionality in reference networks. Self-edges are also discarded. As performance metrics, we used the area under the precision-recall curve (AUPRC) and the area under the receiver operating characteristic (AUROC). For calculations of AUPRC and AUROC, edges of learned graphs are ranked based on the absolute value of their weights and then compared to reference networks.

scSGL returns two adjacency matrices $\mathbf{A}^+$ and $\mathbf{A}^-$, which are used to construct the adjacency matrix of the signed graph. Due to the constraints, $\mathbf{1}^\top \mathbf{A}^+ \mathbf{1} = n$ and $\mathbf{1}^\top \mathbf{A}^- \mathbf{1} = n$, the edge weights for the two adjacencey matrices may not be in the same scale. For instance, if $\mathbf{A}^+$ is denser than $\mathbf{A}^-$, its edge weights are smaller than those of $\mathbf{A}^-$. This can create a problem when ranking the edges for AUROC and AUPRC calculations. To overcome this issue, we construct the adjacency matrix of the signed graph as $\mathbf{A} = \mathbf{A}^+ - s\mathbf{A}^-$, where $\mathbf{A}^-$ is scaled by $s \in \{0.1, 0.2, \ldots, 1, 2, \ldots, 10\}$ and set to the value that gives the best AUPRC value for each dataset.

Hyperparameter Selection: Both the proposed method and the GRN inference techniques require hyperparameter selection. For all methods, hyperparameters were selected separately for each method and each dataset. The hyperparameters were chosen to optimize the AUPRC values with respect to the reference network. We chose to optimise the AUPRC score instead of the AUROC score to account for the inherent sparsity of the true GRN's.

5.1 Simulated Data

The simulated data has been generated using the recently proposed single cell GRN simulator BoolODE [8]. BoolODE converts boolean functions specifying a GRN directly to ODE equations using GeneNetWeaver [43, 44], a widely

Table 1: Statistics for the reference networks of simulated datasets after removing directions and self-edges.

	# Genes	# Pos. Edges	# Neg. Edges
GSD	19	39	20
HSC	11	10	11
mCAD	5	3	6
VSC	8	0	10

used method to simulate bulk transcriptomic data from GRNs. In this paper, we demonstrate the performance of the different methods mentioned above on the curated model simulations illustrated in BoolODE. These datasets are generated from four literature-curated Boolean models: mammalian cortical area development (mCAD), ventral spinal cord (VSC) development, hematopoietic stem cell (HSC) differentiation and gonadal sex determination (GSD). mCAD, VSC, HSC and GSD have varying network densities as reported in Table 1. These datasets represent different types of graph structures, with varying numbers of positive and negative edges; thus serving as good examples for illustrating the robustness of our method in modelling signed graph topologies. To account for the inherent stochasticity of the simulations, BoolODE is used to create ten different synthetic gene expression datasets with 2,000 cells for each model. For each dataset, one version with a dropout rate of 50% and another with a rate of 70% are also considered to observe the performance of the methods under missing values. Since some of the GRN reconstruction methods used for benchmarking require pseudotime information, we calculated the pseudotimes using BEELINE [8]. For evaluation purposes, we only report the mean AUPRC and AUROC scores obtained over the 10 simulations. We refrain from providing the standard deviation estimates for our analysis, as they are almost

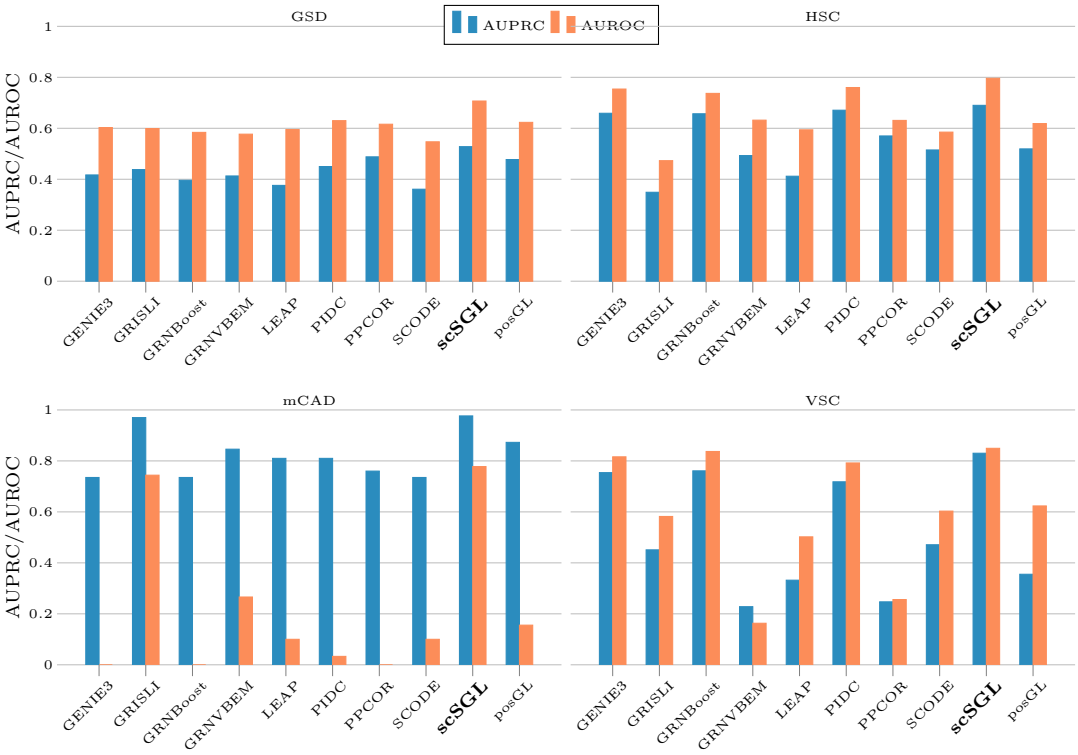

Figure 1: Mean AUPRC and AUROC values for simulated datasets without missing values.

negligible, ranging from 0.001 to 0.03 for all methods considered.

AUPRC and AUROC values of the proposed method along with the considered GRN inference techniques are reported in Figures 1, 2 and 3 for datasets with no missing values, 50% missing values and 70% missing values, respectively. We also report the performance of graph learning when only positive edges are learned, referred to as posGL. For all datasets and missing value cases, scSGL performs better than GRN inference techniques in terms of both AUPRC and AUROC. scSGL also outperforms posGL, which indicates the importance of learning negative edges for GRN inference. This point is especially evident for VSC model, whose reference network contains only negative edges. It can be seen in all figures, posGL performance drops drastically for VSC model. Finally, when Figures 2 and 3 are compared to Figure 1, no drastic change is observed in the performance of any method.

To evaluate the sensitivity of scSGL to hyperparameter selection, we plotted mean AUPRC values for varying α_1 and α_2 values in Figure 4. We report the results only for GSD with no missing values and 50% missing values, but the results are similar for other datasets. It can be observed that the performance of scSGL is not sensitive to changes in hyperparameters as AUPRC values are above 0.5 for most α_1 and α_2 values. This is also the case when there are missing entries in the dataset.

5.2 Experimental scRNAseq

For real scRNA-seq experiments we analyzed two datasets derived from differentiation processes. We used log-transformed transcripts per million reads (TPM) and fragments per millions of kilobases (FPKM) using a pseudocount of 1, as gene expression measurements for the analysis. As the datasets were derived from time-dependendent processes, we averaged the expression of the transcription factors (TF) over all the time points and then chose 100 TF's with the most highly varying average expression. This analysis was carried out in accordance with the

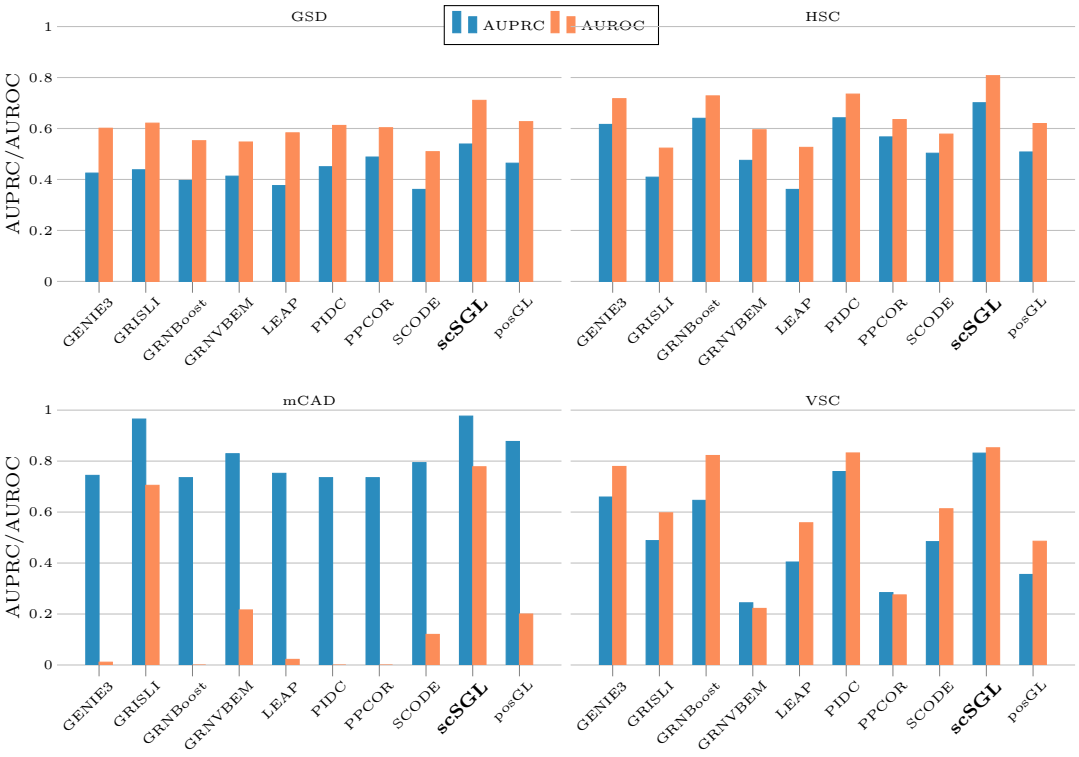

Figure 2: Mean AUPRC and AUROC values for simulated datasets with 50% missing valus.

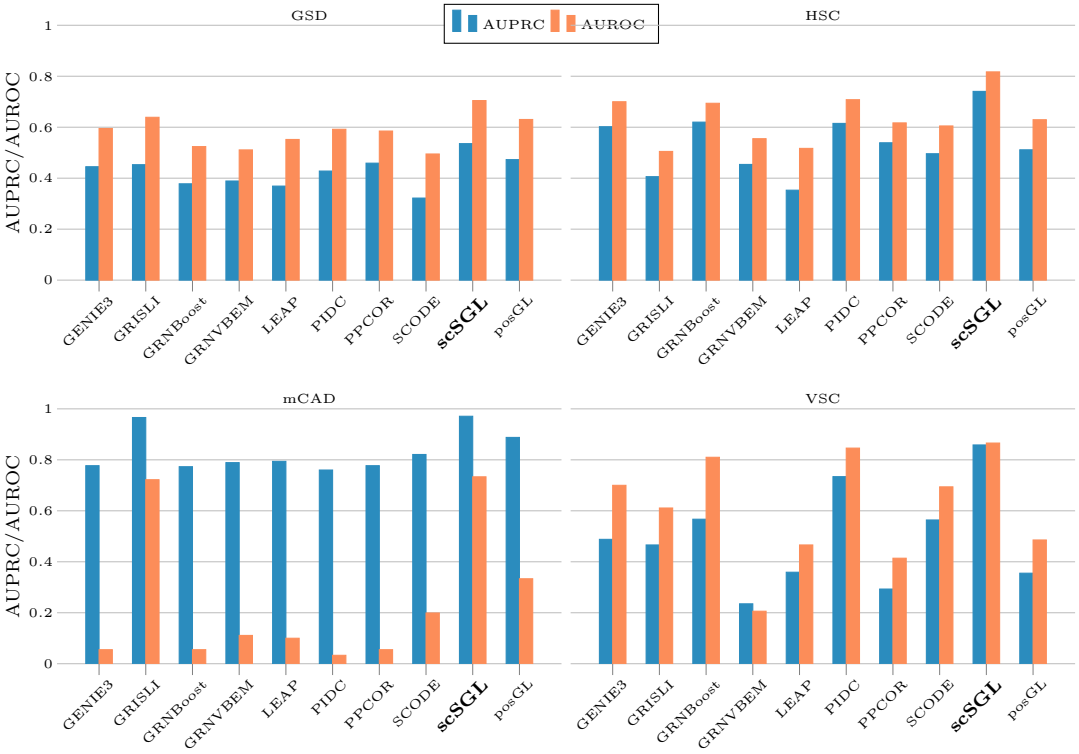

Figure 3: Mean AUPRC and AUROC values for simulated datasets with 70% missing values.

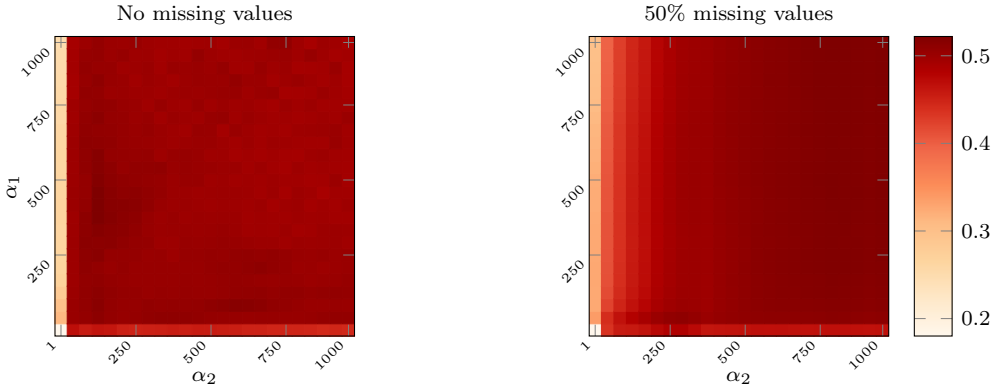

Figure 4: Parameter sensitivity of Mean AUPRC for GSD dataset with no missing values and 50% missing values.

suggestions provided by [19], who first used this data for construction of single cell GRN SCODE.

mESC [45]: This dataset, derived from the differentiation process of mouse embryonic stem cells (mESC) to form primitive endoderm (PrE) cells, contains 356 cells collected at five different time points (0, 12, 24, 48h up to 72h). The pseudotimes for this dataset were computed using Monocle [46] with cells measured at 0 hrs considered to be the root of the differentiation process. This data has been previously used by [19, 7, 20, 25, 8] for single cell GRN evaluation.

hESC [47]: This dataset, derived from the differentiation process of human embryonic stem cells (hESC) to form

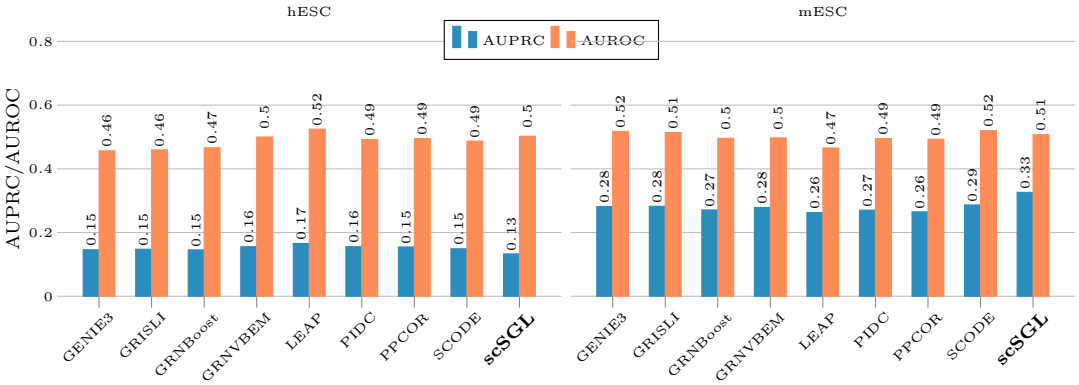

Figure 5: AUPRC plot for real experimental datasets

definitive endoderm cells, contains 758 cells, measured at 0, 12, 24, 36, 72 and 96 hours. The pseudotimes for this dataset were computed using Monocle [46] with cells measured at 0 hrs considered to be the root of the differentiation process. This data has been previously used by [19, 7, 8] for single cell GRN evaluation.

In Figure 5, results for these two datasets are shown. For hESC dataset, six of the methods have AUROC values worse than random predictor, while the remaining methods perform barely above random. In terms of AUPRC, methods have similar values, while scSGL is slightly worse than the other methods. It is worth noting that AUPRC values of the random predictor for hESC is 0.066, so all methods can perform better than random. For mESC dataset, AUROC values of three methods are worse than random, while the remaining are slightly above random. In terms of AUPRC, GRN techniques have similar values, while scSGL outperforms them. Note that AUPRC value of the random estimator for this dataset is 0.102. When results of both datasets are considered together, we observe no method outperforms others consistently across both datasets and both metrics. For instance, the best AUPRC is obtained by LEAP for hESC, while it is obtained by the proposed method for mESC.

When the graphs learned by scSGL are examined in detail, we observed that the best performance is obtained when a sparse network is learned, i.e. the learned graphs have density around 0.005. Moreover, in hESC dataset the best performance of scSGL is obtained when $\alpha_1 = 0$, i.e. only negative edges are learned. However, note that the best performance is defined as the best AUPRC value. This might not be the case for another measure.

6 Conclusion

In this paper, we have introduced a network inference algorithm based on GSP with the purpose of characterizing the gene expression data obtained from cells as graph signals residing on an unknown graph. Our scSGL algorithm identifies functional relationships between genes by learning the signed adjacency matrix from the gene expression data under the assumption that graph signals are similar over positive edges and dissimilar over negative edges. We applied scSGL to four simulated datasets, and two real experimental scRNA-seq datasets during differentiation. For the simulated datasets, scSGL consistently obtained higher AUROC and AUPRC scores in comparison to the benchmarking methods, despite each dataset having a different number of stable cell states. scSGL's superior performance over single cell dynamic GRN methods also points to the fact that pseudotime information may not be crucial for gene regulatory network reconstruction, since all the simulated datasets have multiple pseudotime trajectories, indicative of multiple stable cell states [8]. In addition to that, scSGL's performance remained unaffected for varying dropout rates; across the simulated datasets. (Refer to Figures 1, 2 and 3). We also demonstrated that scSGL attained comparable performance in real data experiments, with the performance of all the GRN reconstruction methods

methods being close to random. Accuracy evaluation of the predicted networks for the real datasets were done using the RikenTFdb and animalTFDB TF-target databases for mouse and human gene expression data, respectively[19]. However, none of these TF-target databases are comprehensive, therefore evaluation using a compilation of reference databases may help in improving the accuracy performance [8].

Although single-cell RNA-seq techniques provide significant advantages over bulk data such as increased sample size with higher depth coverage and and presence of highly distinct cell clusters, it also comes laced with multiple sources of technical and biological noise. Moreover, the inability to differentiate between technical and biological noise, and the absence of adequate noise modelling techniques further exacerbate the problem [48, 49]. scSGL aims to capture the node similarities and dissimilarities based on distances between graph signals. These graph signals exhibit smoothness, which implies that within a given node cluster, genes tend to be homogeneous, while varying across clusters. This leads to densely connected graphs where the heterogeneity induced by distinct cell sub-populations can be simultaneously curbed. Using single cell data with cell cluster labels, easily obtained from single cell clustering algorithms [50], in conjunction with scSGL can aid in identifying functional modules that are associated with a cell type [51]. Integrating pseudotemporal ordering with scSGL can help in identifying the functional modules associated with differential pathways [52].

Despite the availability of a large number of computational methods, accurate GRN reconstruction still remains an open problem. Most reconstruction methods are based on the assumption that presence of an edge implies regulatory relationships. They also have the tendency to establish links between genes regulated by the same regulator. These issues can generate a lot of false positives and therefore additional sources of data such as ChIP-seq measurements that help in identifying direct interactions between TFs and target genes, can provide a way to filter out the spurious interactions [21]. Finally, gene regulation has multiple layers beyond direct TF-target interaction, but functional relationships can only be established if these relationships induce persistent changes in transcriptional state. As single cell data sources over multiple modalities continue to become available, it will be interesting to see how integration of these datatypes aids GRN reconstruction using scSGL. [53].

References

[1] N. Moris, C. Pina, and A. M. Arias, "Transition states and cell fate decisions in epigenetic landscapes," *Nature Reviews Genetics*, vol. 17, no. 11, pp. 693–703, 2016.

[2] M. W. Fiers, L. Minnoye, S. Aibar, C. Bravo González-Blas, Z. Kalender Atak, and S. Aerts, "Mapping gene regulatory networks from single-cell omics data," *Briefings in functional genomics*, vol. 17, no. 4, pp. 246–254, 2018.

[3] A. Saadatpour, G. Guo, S. H. Orkin, and G.-C. Yuan, "Characterizing heterogeneity in leukemic cells using single-cell gene expression analysis," *Genome biology*, vol. 15, no. 12, pp. 1–13, 2014.

[4] V. Moignard, S. Woodhouse, L. Haghverdi, A. J. Lilly, Y. Tanaka, A. C. Wilkinson, F. Buettner, I. C. Macaulay, W. Jawaid, E. Diamanti, *et al.*, "Decoding the regulatory network of early blood development from single-cell gene expression measurements," *Nature biotechnology*, vol. 33, no. 3, pp. 269–276, 2015.

[5] D. Marbach, J. C. Costello, R. Küffner, N. M. Vega, R. J. Prill, D. M. Camacho, K. R. Allison, M. Kellis, J. J. Collins, and G. Stolovitzky, "Wisdom of crowds for robust gene network inference," *Nature methods*, vol. 9, no. 8, pp. 796–804, 2012.

[6] 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.

[7] S. Chen and J. C. Mar, "Evaluating methods of inferring gene regulatory networks highlights their lack of performance for single cell gene expression data," *BMC bioinformatics*, vol. 19, no. 1, pp. 1–21, 2018.

[8] 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.

[9] P. V. Kharchenko, L. Silberstein, and D. T. Scadden, "Bayesian approach to single-cell differential expression analysis," *Nature methods*, vol. 11, no. 7, pp. 740–742, 2014.

[10] G. Finak, A. McDavid, M. Yajima, J. Deng, V. Gersuk, A. K. Shalek, C. K. Slichter, H. W. Miller, M. J. McElrath, M. Prlic, *et al.*, "Mast: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell rna sequencing data," *Genome biology*, vol. 16, no. 1, pp. 1–13, 2015.

[11] D. Risso, F. Perraudeau, S. Gribkova, S. Dudoit, and J.-P. Vert, "Zinb-wave: A general and flexible method for signal extraction from single-cell rna-seq data," *BioRxiv*, p. 125112, 2017.

[12] V. Svensson, "Droplet scrna-seq is not zero-inflated," *Nature Biotechnology*, vol. 38, no. 2, pp. 147–150, 2020.

[13] J. D. Silverman, K. Roche, S. Mukherjee, and L. A. David, "Naught all zeros in sequence count data are the same," *Computational and structural biotechnology journal*, vol. 18, p. 2789, 2020.

[14] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[15] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Learning graphs from data: A signal representation perspective," *IEEE Signal Processing Magazine*, vol. 36, no. 3, pp. 44–63, 2019.

[16] G. Mateos, S. Segarra, A. G. Marques, and A. Ribeiro, "Connecting the dots: Identifying network structure via graph signal processing," *IEEE Signal Processing Magazine*, vol. 36, no. 3, pp. 16–43, 2019.

[17] C. Y. Lim, H. Wang, S. Woodhouse, N. Piterman, L. Wernisch, J. Fisher, and B. Göttgens, "Btr: training asynchronous boolean models using single-cell expression data," *BMC bioinformatics*, vol. 17, no. 1, pp. 1–18, 2016.

[18] S. Woodhouse, N. Piterman, C. M. Wintersteiger, B. Göttgens, and J. Fisher, "Scns: a graphical tool for reconstructing executable regulatory networks from single-cell genomic data," *BMC systems biology*, vol. 12, no. 1, pp. 1–7, 2018.

[19] H. Matsumoto, H. Kiryu, C. Furusawa, M. S. Ko, S. B. Ko, N. Gouda, T. Hayashi, and I. Nikaido, "Scode: an efficient regulatory network inference algorithm from single-cell rna-seq during differentiation," *Bioinformatics*, vol. 33, no. 15, pp. 2314–2321, 2017.

[20] P.-C. Aubin-Frankowski and J.-P. Vert, "Gene regulation inference from single-cell rna-seq data with linear differential equations and velocity inference," *Bioinformatics*, vol. 36, no. 18, pp. 4774–4780, 2020.

[21] S. Aibar, C. B. González-Blas, T. Moerman, H. Imrichova, G. Hulselmans, F. Rambow, J.-C. Marine, P. Geurts, J. Aerts, J. van den Oord, *et al.*, "Scenic: single-cell regulatory network inference and clustering," *Nature methods*, vol. 14, no. 11, pp. 1083–1086, 2017.

[22] 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, pp. 1–10, 2010.

[23] T. E. Chan, M. P. Stumpf, and A. C. Babtie, "Gene regulatory network inference from single-cell data using multivariate information measures," *Cell systems*, vol. 5, no. 3, pp. 251–267, 2017.

[24] A. T. Specht and J. Li, "Leap: constructing gene co-expression networks for single-cell rna-sequencing data using pseudotime ordering," *Bioinformatics*, vol. 33, no. 5, pp. 764–766, 2017.

[25] A. Deshpande, L.-F. Chu, R. Stewart, and A. Gitter, "Network inference with granger causality ensembles on single-cell transcriptomic data," *BioRxiv*, p. 534834, 2019.

[26] N. Papili Gao, S. M. Ud-Dean, O. Gandrillon, and R. Gunawan, "Sincerities: inferring gene regulatory networks from time-stamped single cell transcriptional expression profiles," *Bioinformatics*, vol. 34, no. 2, pp. 258–266, 2018.

[27] M. Sanchez-Castillo, D. Blanco, I. M. Tienda-Luna, M. Carrion, and Y. Huang, "A bayesian framework for the inference of gene regulatory networks from time and pseudo-time series data," *Bioinformatics*, vol. 34, no. 6, pp. 964–970, 2018.

[28] S. Chen, A. Sandryhaila, J. M. Moura, and J. Kovačević, "Signal recovery on graphs: Variation minimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 17, pp. 4609–4624, 2015.

[29] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning laplacian matrix in smooth graph signal representations," *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6160–6173, 2016.

[30] V. Kalofolias, "How to learn a graph from smooth signals," in *Artificial Intelligence and Statistics*, pp. 920–929, 2016.

[31] J. Hou, L.-P. Chau, Y. He, and H. Zeng, "Robust laplacian matrix learning for smooth graph signals," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 1878–1882, IEEE, 2016.

[32] P. Berger, G. Hannak, and G. Matz, "Efficient graph learning from noisy and incomplete data," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, pp. 105–119, 2020.

[33] S. K. Kadambari and S. P. Chepuri, "Learning product graphs from multidomain signals," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5665–5669, IEEE, 2020.

[34] B. Le Bars, P. Humbert, L. Oudre, and A. Kalogeratos, "Learning laplacian matrix from bandlimited graph signals," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2937–2941, IEEE, 2019.

[35] L. Rui, H. Nejati, S. H. Safavi, and N.-M. Cheung, "Simultaneous low-rank component and graph estimation for high-dimensional graph signals: Application to brain imaging," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4134–4138, IEEE, 2017.

[36] G. Matz and T. Dittrich, "Learning signed graphs from data," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5570–5574, IEEE, 2020.

[37] J. Kunegis, S. Schmidt, A. Lommatzsch, J. Lerner, E. W. De Luca, and S. Albayrak, "Spectral analysis of signed graphs for clustering, prediction and visualization," in *Proceedings of the 2010 SIAM International Conference on Data Mining*, pp. 559–570, SIAM, 2010.

[38] V. A. Traag and J. Bruggeman, "Community detection in networks with positive and negative links," *Physical Review E*, vol. 80, no. 3, p. 036115, 2009.

[39] H. Scheel and S. Scholtes, "Mathematical programs with complementarity constraints: Stationarity, optimality, and sensitivity," *Mathematics of Operations Research*, vol. 25, no. 1, pp. 1–22, 2000.

[40] Y. Wang, W. Yin, and J. Zeng, "Global convergence of admm in nonconvex nonsmooth optimization," *Journal of Scientific Computing*, vol. 78, no. 1, pp. 29–63, 2019.

[41] T. Moerman, S. Aibar Santos, C. Bravo González-Blas, J. Simm, Y. Moreau, J. Aerts, and S. Aerts, "Grnboost2 and arboreto: efficient and scalable inference of gene regulatory networks," *Bioinformatics*, vol. 35, no. 12, pp. 2159–2161, 2019.

[42] S. Kim, "ppcor: an r package for a fast calculation to semi-partial correlation coefficients," *Communications for statistical applications and methods*, vol. 22, no. 6, p. 665, 2015.

[43] T. Schaffter, D. Marbach, and D. Floreano, "Genenetweaver: in silico benchmark generation and performance profiling of network inference methods," *Bioinformatics*, vol. 27, no. 16, pp. 2263–2270, 2011.

[44] D. Marbach, T. Schaffter, C. Mattiussi, and D. Floreano, "Generating realistic in silico gene networks for performance assessment of reverse engineering methods," *Journal of computational biology*, vol. 16, no. 2, pp. 229–239, 2009.

[45] 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, pp. 1–16, 2018.

[46] C. Trapnell, D. Cacchiarelli, J. Grimsby, P. Pokharel, S. Li, M. Morse, N. J. Lennon, K. J. Livak, T. S. Mikkelsen, and J. L. Rinn, "The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells," *Nature biotechnology*, vol. 32, no. 4, p. 381, 2014.

[47] 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, no. 1, pp. 1–20, 2016.

[48] D. Grün, L. Kester, and A. Van Oudenaarden, "Validation of noise models for single-cell transcriptomics," *Nature methods*, vol. 11, no. 6, pp. 637–640, 2014.

[49] O. Stegle, S. A. Teichmann, and J. C. Marioni, "Computational and analytical challenges in single-cell transcriptomics," *Nature Reviews Genetics*, vol. 16, no. 3, pp. 133–145, 2015.

[50] R. Petegrosso, Z. Li, and R. Kuang, "Machine learning and statistical methods for clustering single-cell rna-sequencing data," *Briefings in bioinformatics*, vol. 21, no. 4, pp. 1209–1223, 2020.

[51] Z. Wang, M. Cui, A. M. Shah, W. Tan, N. Liu, R. Bassel-Duby, and E. N. Olson, "Cell-type-specific gene regulatory networks underlying murine neonatal heart regeneration at single-cell resolution," *Cell reports*, vol. 33, no. 10, p. 108472, 2020.

[52] Z. Xue, K. Huang, C. Cai, L. Cai, C.-y. Jiang, Y. Feng, Z. Liu, Q. Zeng, L. Cheng, Y. E. Sun, *et al.*, "Genetic programs in human and mouse early embryos revealed by single-cell rna sequencing," *Nature*, vol. 500, no. 7464, pp. 593–597, 2013.

[53] T. Stuart, A. Butler, P. Hoffman, C. Hafemeister, E. Papalexi, W. M. Mauck III, Y. Hao, M. Stoeckius, P. Smibert, and R. Satija, "Comprehensive integration of single-cell data," *Cell*, vol. 177, no. 7, pp. 1888–1902, 2019.