

GRNITE: Gene Regulatory Network Inference with Text Embeddings

Ali Azizpour^{1,*}, Narein Rao^{2,*}[0009-0005-0495-3671], Santiago Segarra^{1,4}[0000-0002-8408-9633], Luay Nakhleh^{2,3,4}[0000-0003-3288-6769], and Nicolae Sapoval^{2,†}[0000-0002-0736-5075]

¹ Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005

² Department of Computer Science, Rice University, Houston, TX 77005

³ Department of BioSciences, Rice University, Houston, TX 77005

⁴ Ken Kennedy Institute, Rice University, Houston, TX 77005

* These authors have contributed equally to this work.

† Corresponding author: nsapoval@rice.edu

Abstract. Gene regulatory networks (GRNs) capture complex regulatory relationships that govern gene expression in cells. Inference of GRNs from single-cell RNA-seq (scRNA-seq) data has been an active topic of research in the past several years. However, despite the improvements in the data quality, the GRN inference problem remains a challenging task with many approaches showing variable performance dependent on the organism and cell type. To improve the quality of GRN inference and enable more comprehensive exploratory analyses of GRNs across various phenotypes of interest, we developed a two-stage meta-method called GRNITE. In the first step, GRNITE leverages LLM-based embeddings of plain text gene descriptions to create a prior gene interaction graph, which is then optimized with a graph neural network (GNN) to achieve a “universal” biological prior for GRN inference. In the second step, GRNITE uses a GNN to incorporate information from a GRN inferred from scRNA-seq data with any baseline inference method into our prior. The result of this two-step approach is a near-universal improvement in AUROC and recall of all evaluated methods, with minor trade-offs in precision. Furthermore, GRNITE is a lightweight meta-method, which adds a minimal amount of extra compute time on top of the original GRN inference performed. GRNITE and our pre-trained universal prior GRN are available on GitHub: <https://github.com/aliaz99/GRNITE>.

Keywords: Gene regulatory networks · Graph neural networks · Text embeddings

1 Introduction

Gene regulatory networks (GRNs) represent complex molecular interactions that govern gene expression within cells. Inference of GRNs from gene expression data is a longstanding problem in computational biology with multiple methods developed over the past two decades [11,12,22,8,23]. With the emergence of single-cell sequencing technologies, methods for GRN inference from these data began to be developed. However, due to current technological limitations, single-cell RNA sequencing (scRNA-seq) data is prone to gene dropout [38]. This challenge encouraged the development of methods that can explicitly account for the noisy nature of data, as well as the development of multi-omic methods that aim to leverage multiple data modalities (e.g., scRNA-seq + scATAC-seq) to improve the resolution and robustness of GRN inference [31,19,21].

Recently, deep neural networks, such as convolutional neural networks (CNNs) and variational autoencoders (VAEs), have been applied to scRNA-seq data to model complex, non-linear regulatory relationships [32,6,25]. While effective, these methods mostly focus on pairwise transcription factor (TF)–target interactions and often overlook the global regulatory structure of GRNs. Graph neural networks (GNNs) [29] offer a powerful alternative by explicitly modeling genes as nodes and regulatory relationships as edges. GNNs have been successfully used in diverse domains such as social [30,15,24], biological [18,3,17,5,28], and wireless communication networks [34,35] to model structured interactions between entities, enabling tasks such as node classification, community detection, and link prediction [9]. Prior work [27,16] has demonstrated that casting GRN inference as a link prediction problem with a graph autoencoder allows effective integration of global regulatory structure and improves robustness to noise in single-cell data.

Notably, in parallel to these developments, databases of ChIP-seq-confirmed TF-target interactions were expanded, allowing for the construction of stronger evidence-based priors for GRN inference [20,36,2]. These

enriched resources have supported the emergence of prior-augmented GRN inference frameworks. Despite these advances, there remains a need for robust and reliable GRN inference approaches that can be applied in exploratory studies with limited references [26]. With the expanded knowledge of individual gene functions, databases describing those in natural language also grew [4]. Additionally, with the emergence of large language models (LLMs) as powerful zero-shot knowledge extractors for natural language, automated knowledge graph construction from plain text became widely accessible [7,37].

Based on a recent benchmarking study of the effectiveness of various pretrained gene embeddings for downstream biological tasks [37], GenePT [7] was identified as a high-quality embedding model for gene-level features. This finding suggests that gene embeddings derived from plain text descriptions can be leveraged as useful priors for gene-based biological inference tasks. Thus, our goal in this study is two-fold: (1) we evaluate a hypothesis that text-based gene embeddings can be used as an effective prior for GRN inference, and (2) we develop GRNITE, a GNN-based tool that can be used in conjunction with any scRNA-seq GRN inference software in order to gain improved performance for GRN inference.

In particular, our framework for **GRN Inference with Text Embeddings** (GRNITE) leverages text modality by replacing co-expression priors with a text-informed prior derived from gene descriptions. This prior is then denoised and refined using curated TF–target interactions to form a biologically grounded scaffold. Moreover, rather than proposing another standalone GRN inference algorithm, we design GRNITE as a flexible pipeline: our enhanced prior is integrated with the output of any existing GRN inference method, allowing us to systematically improve performance and recover biologically plausible interactions without replacing the underlying method itself.

Our results demonstrate that GRNITE increases the Jaccard similarity between the inferred and ground-truth edge sets on benchmarking data. Additionally, we show improvement in balanced accuracy and a universal increase in recall across benchmarking datasets. Furthermore, GRNITE increases inter-method consensus and improves the accuracy of consensus edge sets. Finally, we demonstrate that on biological data, GRNITE enables the superior recovery of key TF-target interactions.

2 Methods

In this section, we present **GRNITE**, our method for enhanced GRN inference using text-derived embeddings. An overview of our three-step framework is shown in Figure 1. In Section 2.1, we describe how to construct a text-informed prior graph. Next, in Section 2.2, we apply a graph autoencoder to further refine this prior using additional biological knowledge into a unified prior graph. In Section 2.3, we design a pipeline that leverages this improved prior to enhance the output of any arbitrary GRN inference method. Finally, in Section 2.4, we describe the evaluation setup, including datasets, metrics, and experimental procedures.

2.1 Text-informed Prior Construction

We begin by constructing a prior gene interaction graph in which connections reflect similarities in text-derived information. To achieve this, we first obtain a text-based embedding vector for each gene by following a procedure similar to the GenePT approach [7]. Specifically, we first obtain gene information and text summaries from the NCBI Gene database. Then we filter out genes that do not have a matching taxonomic ID for our dataset of interest, and any genes that have an empty gene symbol. Finally, the “Summary” field containing brief summary description of the gene, its function (whenever known) and products in plain-text English [4], from the downloaded database is used to compute gene embeddings with Qwen3 embedding 8B model [33]. This results in a $\mathbf{X}_{\text{emb}} \in \mathbb{R}^{G \times F}$ matrix of gene embeddings where G is the number of genes, and $F = 4096$ by default.

Now, let $\mathbf{x}_i = \mathbf{X}_{\text{emb}}(i, :) \in \mathbb{R}^F$ denote the embedding of gene i . Using these embeddings, we compute a weighted similarity matrix $\mathbf{W}_{\text{emb}} \in \mathbb{R}^{G \times G}$ by measuring the cosine similarity between pairs of genes:

$$\mathbf{W}_{\text{emb}}(i, j) = 1 - \frac{\mathbf{x}_i^\top \mathbf{x}_j}{\|\mathbf{x}_i\|_2 \|\mathbf{x}_j\|_2}, \quad \forall i, j \in \{1, \dots, G\},$$

This matrix quantifies how close two genes are in the text-embedding space, with smaller values indicating greater similarity.

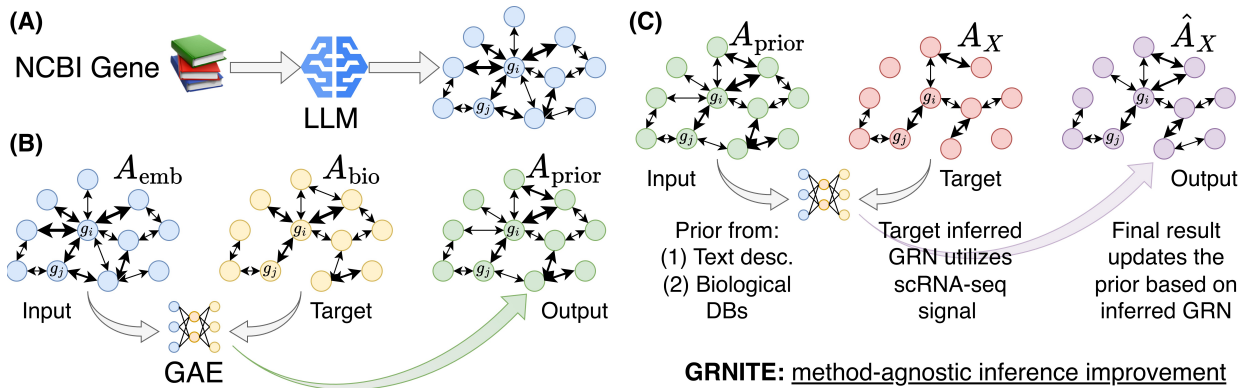


Fig. 1: Overview of the GRNITE framework. (A) Embeddings for gene descriptions from NCBI Gene database are computed with a large language model (LLM), and cosine similarity between the resulting vectors is used to construct an embedding prior adjacency matrix ($\mathbf{A}_{\text{emb}}$). (B) A graph autoencoder (GAE) model is trained to reconstruct a biological prior graph ($\mathbf{A}_{\text{bio}}$, based on a database of known TF-target interactions) from $\mathbf{A}_{\text{emb}}$. The final output is an updated prior graph $\mathbf{A}_{\text{prior}}$ that fuses information from text embeddings and known interactions. (C) At inference time, a GRN $\mathbf{A}_X$ is inferred with an arbitrary method from scRNA-seq gene expression data. Then a GAE is trained to reconstruct $\mathbf{A}_X$ from $\mathbf{A}_{\text{prior}}$, and the final prediction $\hat{\mathbf{A}}_X$ is produced.

Next, we convert this weighted similarity matrix into a binary adjacency matrix $\mathbf{A}_{\text{emb}} \in \{0, 1\}^{G \times G}$. To do so, we threshold the distances using the average pairwise distance across all gene pairs:

$$\mathbf{A}_{\text{emb}}(i, j) = \begin{cases} 1, & \text{if } \mathbf{W}_{\text{emb}}(i, j) \leq \tau_1, \\ 0, & \text{otherwise,} \end{cases} \quad \forall i, j \in \{1, \dots, G\},$$

where τ_1 is a tunable hyperparameter. This retains only the most similar gene pairs, forming a sparse prior graph that captures strong relationships based on textual information.

2.2 Biology-informed Prior Enhancement

In this step, our goal is to integrate biological knowledge, such as experimentally validated TF-target interactions, into the text-derived graph $\mathbf{A}_{\text{emb}}$ to construct a refined and biologically meaningful prior graph. Let $\mathbf{A}_{\text{bio}} \in \{0, 1\}^{G \times G}$ denote the biological prior adjacency matrix, where $\mathbf{A}_{\text{bio}}(i, j) = 1$ if a known interaction between genes i and j exists in curated databases. Our objective is to learn a new prior graph $\mathbf{A}_{\text{prior}}$ that combines the information in $\mathbf{A}_{\text{emb}}$ and $\mathbf{A}_{\text{bio}}$, i.e., combines dense text-based similarities with high-confidence biological interactions.

To achieve this, we employ a graph autoencoder (GAE) that takes $\mathbf{A}_{\text{emb}}$ as the structural input and uses $\mathbf{X}_{\text{exp}}$ as node features, where $\mathbf{X}_{\text{exp}} \in \mathbb{R}^{G \times C}$ is the gene expression matrix with C denoting the number of cells. Formally, we denote the encoder as a function $g_{\theta_1}(\cdot)$ parameterized by θ_1 and the decoder as $h_{\theta_2}(\cdot)$ parameterized by θ_2 . The encoder maps node features into a latent representation:

$$\mathbf{Z} = g_{\theta_1}(\mathbf{X}_{\text{exp}}, \mathbf{A}_{\text{emb}}) \in \mathbb{R}^{G \times d}.$$

In practice, g_{θ_1} consists of multiple topology-aware graph convolution (TAGConv) layers [10]. A single TAGConv layer is defined as:

$$\mathbf{H}^{(\ell+1)} = \sum_{k=0}^K \left(\tilde{\mathbf{D}}^{-1/2} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-1/2} \right)^k \mathbf{H}^{(\ell)} \mathbf{W}_k,$$

where $\tilde{\mathbf{A}} = \mathbf{A}_{\text{emb}}$, $\tilde{\mathbf{D}}$ is its degree matrix, $\mathbf{H}^{(\ell)}$ is the input feature matrix at layer ℓ , $\mathbf{W}_k$ are learnable weights, and K is the number of hops from which the data is aggregated. Unlike standard GCN layers [14]

that aggregate information only from one-hop neighbors ($K = 1$), TAGConv aggregates information from multi-hop neighborhoods up to K hops, enabling richer propagation of topological information and better capturing long-range dependencies in the prior graph. The encoder initializes with $\mathbf{H}^{(0)} = \mathbf{X}_{\text{exp}}$ and produces latent representations $\mathbf{Z} = \mathbf{H}^{(L)} \in \mathbb{R}^{G \times d}$ after L layers.

Given the latent embeddings $\mathbf{z}_i$ for each gene, the decoder reconstructs an edge score using a bilinear form:

$$\hat{\mathbf{A}}(i, j) = h_{\theta_2}(\mathbf{z}_i, \mathbf{z}_j) = \mathbf{z}_i^\top \mathbf{W} \mathbf{z}_j,$$

where $\mathbf{W} \in \mathbb{R}^{d \times d}$ is a learnable weight matrix.

To train the model, we compare the reconstructed scores $\hat{\mathbf{A}}$ against the biological prior adjacency matrix $\mathbf{A}_{\text{bio}}$. However, since $\mathbf{A}_{\text{bio}}$ is highly sparse, directly evaluating the loss over all G^2 possible pairs would lead to severe class imbalance and redundant computations. Therefore, we perform *negative sampling*: we include all positive edges (where $\mathbf{A}_{\text{bio}}(i, j) = 1$) and a randomly sample a subset of negative edges (where $\mathbf{A}_{\text{bio}}(i, j) = 0$) from the upper triangular part of the matrix. Let Ω denote the set of sampled edge pairs.

The reconstruction loss is then defined as:

$$\mathcal{L}_{\text{bio}} = -\frac{1}{|\Omega|} \sum_{(i, j) \in \Omega} \left[\mathbf{A}_{\text{bio}}(i, j) \log \sigma(\hat{\mathbf{A}}(i, j)) + (1 - \mathbf{A}_{\text{bio}}(i, j)) \log (1 - \sigma(\hat{\mathbf{A}}(i, j))) \right],$$

where $\sigma(\cdot)$ denotes the sigmoid activation applied to the raw predicted interaction logits. A positive weighting term can optionally be applied to further mitigate class imbalance by emphasizing true interactions.

The optimal parameters are obtained by minimizing this loss:

$$\theta^* = \arg \min_{\theta} \mathcal{L}_{\text{bio}}, \quad \theta = \theta_1 \cup \theta_2.$$

After training, the final reconstructed edge probabilities are obtained as:

$$\hat{\mathbf{A}}_s(i, j) = \sigma(h_{\theta_2}(g_{\theta_1}(\mathbf{X}_{\text{exp}}, \mathbf{A}_{\text{emb}}))(i, j)), \quad \forall i, j \in \{1, \dots, G\}.$$

Finally, the refined prior graph is constructed by thresholding the reconstructed scores:

$$\mathbf{A}_{\text{prior}}(i, j) = \mathbf{1}\{\hat{\mathbf{A}}_s(i, j) \geq \tau_2\}, \quad \tau_2 \in (0, 1),$$

where τ_2 is a tunable threshold. In this way, we first learn the optimal encoder and decoder parameters using $\mathbf{A}_{\text{bio}}$ as supervision, and then use the trained model to produce refined edge scores that are subsequently binarized to construct the final prior graph.

This procedure retains biologically supported edges while preserving additional plausible interactions inferred from text. Since the encoder operates on $\mathbf{A}_{\text{emb}}$, genes connected in the text-derived graph naturally acquire similar embeddings; however, by supervising reconstruction using $\mathbf{A}_{\text{bio}}$, the model is encouraged to suppress edges that are inconsistent with biological evidence, effectively denoising and refining the text-based prior.

2.3 Integration with Existing GRN Inference Methods

In the final step, we incorporate the enhanced prior graph obtained from the previous stage to improve the output of any arbitrary GRN inference method. Since this prior embeds complementary information from both biological databases and text-derived embeddings, our goal is to leverage it in a way that enhances, rather than replaces, existing GRN inference pipelines.

Let $\mathbf{A}_X \in [0, 1]^{G \times G}$ represent the output of an arbitrary GRN inference method X , obtained from the gene expression data $\mathbf{X}_{\text{exp}}$. We seek to refine this output graph by incorporating the rich contextual knowledge captured in the prior graph $\mathbf{A}_{\text{prior}}$.

To achieve this, we employ a graph autoencoder similar to the previous step.

$$\mathbf{Z} = g_{\psi_1}(\mathbf{X}_{\text{rnd}}, \mathbf{A}_{\text{prior}}), \quad \hat{\mathbf{A}}(i, j) = h_{\psi_2}(\mathbf{z}_i, \mathbf{z}_j) = \mathbf{z}_i^\top \mathbf{W} \mathbf{z}_j, \quad \forall i, j \in \{1, \dots, G\},$$

where $\mathbf{Z} \in \mathbb{R}^{G \times d}$ is the latent gene representation, $\mathbf{W} \in \mathbb{R}^{d \times d}$ is a learnable bilinear matrix. Unlike the previous step, where expression data $\mathbf{X}_{\text{exp}}$ was used as node features, here we use random Gaussian features

$\mathbf{X}_{\text{rnd}}$ to prevent information leakage from the target method $\mathbf{A}_X$, which is already inferred from expression data [1]. The encoder is constructed using TAGConv layers, which enable multi-hop message passing and effectively capture long-range dependencies encoded in $\mathbf{A}_{\text{prior}}$.

Different from the previous step, the training objective here encourages the reconstructed graph $\hat{\mathbf{A}}$ to remain consistent with both the target graph $\mathbf{A}_X$ and the enhanced prior $\mathbf{A}_{\text{prior}}$. To accomplish this, we again employ negative sampling: we include all positive edges (where $\mathbf{A}_X(i, j) = 1$) and a randomly sampled subset of negative edges (where $\mathbf{A}_X(i, j) = 0$) from the upper triangular part of the matrix to avoid duplication. Let Ω denote this set of sampled edge pairs.

The two binary cross-entropy with logits losses are defined as:

$$\mathcal{L}_{\text{target}} = -\frac{1}{|\Omega|} \sum_{(i,j) \in \Omega} \left[\mathbf{A}_X(i, j) \log \sigma(\hat{\mathbf{A}}(i, j)) + (1 - \mathbf{A}_X(i, j)) \log(1 - \sigma(\hat{\mathbf{A}}(i, j))) \right],$$

$$\mathcal{L}_{\text{prior}} = -\frac{1}{|\Omega|} \sum_{(i,j) \in \Omega} \left[\mathbf{A}_{\text{prior}}(i, j) \log \sigma(\hat{\mathbf{A}}(i, j)) + (1 - \mathbf{A}_{\text{prior}}(i, j)) \log(1 - \sigma(\hat{\mathbf{A}}(i, j))) \right].$$

The total loss combines these two objectives as:

$$\mathcal{L}_{\text{total}} = \beta \mathcal{L}_{\text{target}} + (1 - \beta) \mathcal{L}_{\text{prior}}, \quad \beta \in [0, 1],$$

where β balances the emphasis between maintaining fidelity to the target graph and preserving similarity to the enhanced prior. The optimal parameters are obtained as:

$$\Psi^* = \underset{\Psi}{\operatorname{argmin}} \quad \mathcal{L}_{\text{total}}, \quad \Psi = \psi_1 \cup \psi_2.$$

Once trained, the final reconstructed scores are computed using the optimized model:

$$\hat{\mathbf{A}}_f(i, j) = \sigma(h_{\psi_2^*}(g_{\psi_1^*}(\mathbf{X}_{\text{rnd}}, \mathbf{A}_{\text{prior}}))(i, j)), \quad \forall i, j \in \{1, \dots, G\}.$$

Finally, we threshold these reconstructed scores to obtain the refined GRN:

$$\mathbf{A}_{\text{output}}(i, j) = \mathbf{1}\{\hat{\mathbf{A}}_f(i, j) \geq \tau_3\}, \quad \tau_3 \in (0, 1).$$

This formulation allows GRNITE to remain fully agnostic to the underlying GRN inference algorithm. The encoder-decoder operates over the prior graph $\mathbf{A}_{\text{prior}}$, ensuring that the latent representations already incorporate biologically grounded and text-informed relationships, while the loss function encourages consistency with the chosen target method. Consequently, this approach enhances the target graph by pruning weak or inconsistent edges and, more importantly, by recovering biologically plausible edges that the base method might have missed. Since the model is independent of the specific choice of $\mathbf{X}$, any GRN inference method can serve as the target, allowing flexible and systematic improvement of existing GRN pipelines.

2.4 Datasets and Evaluation

Datasets We leveraged commonly used benchmarking datasets for GRN inference evaluation. In particular, we employed two human and five mouse datasets from BEELINE considering both TFs and 500 and 1000 highly variable genes (HVGs), referred to as BEELINE TF+500 and BEELINE TF+1000 data. We also used simulated data from GRouNdGAN [39]. For BEELINE we relied on accompanying ChIP-seq data to provide ground truth reference for interactions, while for GRouNdGAN we used the ground truth GRN used for the simulation. For the analysis of biological single-cell data, we used two datasets, a high-fat-diet mouse pancreatic β -cell dataset (HFD) and peripheral blood mononuclear cell dataset (PBMC). The high-fat-diet mouse pancreatic β -cell dataset is divided into two subsets: cells with high *Cd81* expression and cells with low *Cd81* expression. Four immune cell-type specific datasets were also derived from the peripheral blood mononuclear cell dataset, namely classical monocytes, myeloid dendritic cells, naive B cells, and naive CD4 T cells. We acquired 27 ground truth driver TF-target interactions for the high-fat-diet mouse dataset following the workflow provided in KEGNI [16]. We also acquired 1788 TF-target interactions for all immune cell-type specific interactions within the PBMC dataset curated from top 100 TF-target interactions available via cistromeDB [20,36].

Evaluation GRN inference methods are commonly evaluated using metrics such as the Area Under the Receiver Operating Characteristic curve (AUROC) and the Jaccard index. In line with prior work and the BEELINE benchmark [23], we adopt the same evaluation metrics across all datasets to ensure a fair comparison. The evaluation is performed over the entire space of possible TF–target interactions, considering all potential TF–target pairs as candidates for regulatory edges.

Following the BEELINE evaluation setup, the AUROC in our case is equivalent to the balanced accuracy. This is due to the ROC curve being reduced to three operating points: the two trivial cases of predicting all edges as positive or all as negative, and the operating point corresponding to the inferred network. The area under this curve is equal to the average of the *true positive rate* (TPR) and the *true negative rate* (TNR), i.e., $\text{AUROC} = (\text{TPR} + \text{TNR})/2$, where $\text{TPR} = \text{TP}/(\text{TP} + \text{FN})$ measures the proportion of correctly predicted true edges, and $\text{TNR} = \text{TN}/(\text{TN} + \text{FP})$ measures the proportion of correctly predicted non-edges. This interpretation is particularly meaningful for GRNs as true regulatory interactions are sparse, resulting in a strong class imbalance.

In addition to AUROC, we report the Jaccard index to directly measure the overlap between the predicted and reference edge sets. For a predicted edge set $\hat{E}$ and ground truth edge set E , the Jaccard index is defined as: $\text{Jaccard}(\hat{E}, E) = |\hat{E} \cap E|/|\hat{E} \cup E|$. While AUROC quantifies how well positive edges are ranked above negative ones, the Jaccard index evaluates exact edge-set similarity, making it a complementary and stricter measure for GRN assessment.

Finally, since in practice no ground truth is available, and consensus edges between multiple method runs are often used as a proxy for high confidence interactions, we also evaluated Jaccard similarities of edge sets inferred by different methods. We further examined how this overlap in edge predictions is affected by application of GRNITE, and whether the updated edge set intersection offers a better AUROC when compared to the baseline.

3 Results

3.1 Baseline performance

We evaluate the baseline performance of the methods used for inference of the target GRN from the scRNA-seq data on BEELINE and GRouNdGAN datasets (Tab. 1). First, we note that CellOracle has the highest AUROC on BEELINE, and highest Jaccard similarity to true edge sets across all data. On the GRouNdGAN dataset PORTIA achieves the highest AUROC, with CellOracle and GRNBoost being second best in performance. DAZZLE [38] was unable to run on GRouNdGAN simulated data, and larger biological datasets, and hence has been only included in the BEELINE comparisons.

Table 1: Baseline performance of the GRN inference methods measured as the mean ($\pm$ st. dev.) AUROC and edge set Jaccard similarity for the BEELINE and GRouNdGAN datasets. Highest mean value in each column is underlined.

Method	AUROC			Jaccard Sim.		
	BEELINE (500)	BEELINE (1000)	GRouNdGAN	BEELINE (500)	BEELINE (1000)	GRouNdGAN
Correlation	0.495 $\pm$ 0.028	0.498 $\pm$ 0.028	0.516 $\pm$ 0.035	0.015 $\pm$ 0.011	0.011 $\pm$ 0.008	0.019 $\pm$ 0.008
GRNBoost	0.502 $\pm$ 0.017	0.504 $\pm$ 0.011	0.591 $\pm$ 0.066	0.016 $\pm$ 0.009	0.011 $\pm$ 0.007	0.042 $\pm$ 0.019
PORTIA	0.501 $\pm$ 0.012	0.502 $\pm$ 0.005	<u>0.608</u> $\pm$ 0.063	0.015 $\pm$ 0.008	0.010 $\pm$ 0.005	0.068 $\pm$ 0.033
DAZZLE	0.523 $\pm$ 0.021	0.526 $\pm$ 0.025	—	0.017 $\pm$ 0.009	0.012 $\pm$ 0.007	—
CellOracle	<u>0.690</u> $\pm$ 0.048	<u>0.697</u> $\pm$ 0.050	0.593 $\pm$ 0.029	<u>0.112</u> $\pm$ 0.059	<u>0.115</u> $\pm$ 0.063	<u>0.125</u> $\pm$ 0.020
SCENIC	0.510 $\pm$ 0.005	0.506 $\pm$ 0.004	0.514 $\pm$ 0.016	0.019 $\pm$ 0.010	0.012 $\pm$ 0.008	0.027 $\pm$ 0.031

3.2 Improvement in inference performance

BEELINE Datasets We evaluated the improvements in AUROC and Jaccard similarity of the GRN edge sets on 2 human and 5 mice datasets (Fig. 2). For all methods except CellOracle (CO), GRNITE provides a

positive change in the AUROC. For the mDC and mESC datasets, use of GRNITE with CellOracle inferred target graph leads to a minor (< 0.01) decrease in the AUROC.

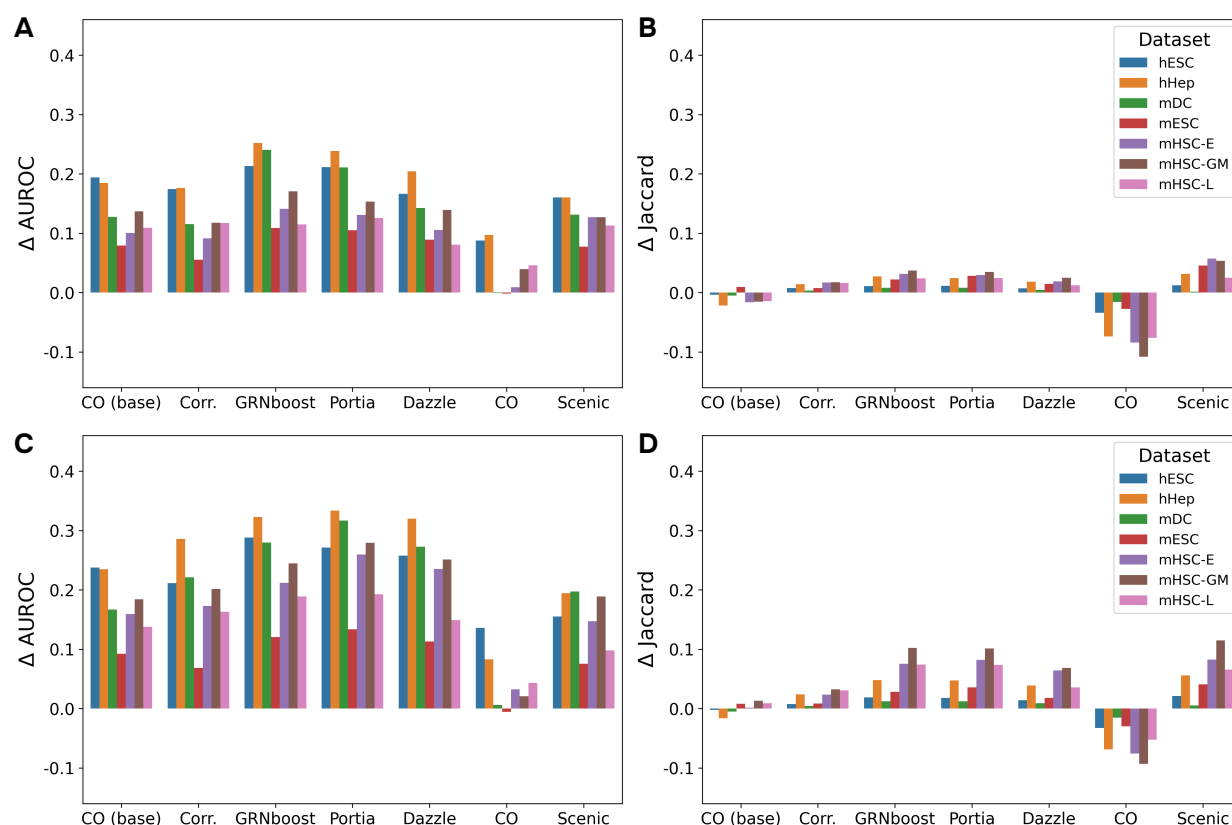


Fig. 2: Change in performance from baseline methods to GRNITE. Change in the area under the receiver operating characteristic curve (Δ AUROC: A, C) and change in the Jaccard similarity of edge sets (Δ Jaccard: B, D) after our method has been applied to the target graph. Panels A and B correspond to BEELINE TF+500, and panels C and D correspond to BEELINE TF+1000 datasets.

Similarly, for all methods except CellOracle, GRNITE provides a moderate positive improvement in the Jaccard similarity of the inferred and true edge sets (Fig. 2). We note that the improvement in both AUROC and Jaccard similarity is higher in magnitude for BEELINE TF+1000 dataset (Fig. 2). For both BEELINE TF+500 and TF+1000 datasets all methods except CellOracle experience a positive change in F1-score, driven primarily by a large increase in recall (Supp. Fig. 8, 9). Notably, more conservative methods (CellOracle, SCENIC) which predict fewer edges, but have higher precision, lose some of their precision after application of GRNITE (Supp. Fig. 8, 9). Finally, ablations with respect to using a purely biological prior ($\mathbf{A}_{\text{bio}}$, Supp. Fig. 10, 11, 14, 15) or purely text embedding-based prior ($\mathbf{A}_{\text{emb}}$, Supp. Fig. 12, 13, 16, 17) show better performance from the combined prior across the majority of metrics.

GRouNdGAN datasets For the GRouNdGAN simulated data, we observe a pattern of performance changes similar to the one on BEELINE. Specifically, all methods receive an improvement in AUROC, and CellOracle is the only method with consistently decreased Jaccard similarity for edge sets (Fig. 3).

Notably, Portia also shows reduced Jaccard edge set similarity on the PBMC-CTL and Bone Marrow datasets. This matches the observation that these are the datasets on which GRNITE + Portia also shows a small reduction in precision (Supp. Fig. 18). Similarly to BEELINE, we observe a universal improvement in recall, and an improvement in F1-score for all the methods except CellOracle (Supp. Fig. 18). However, on

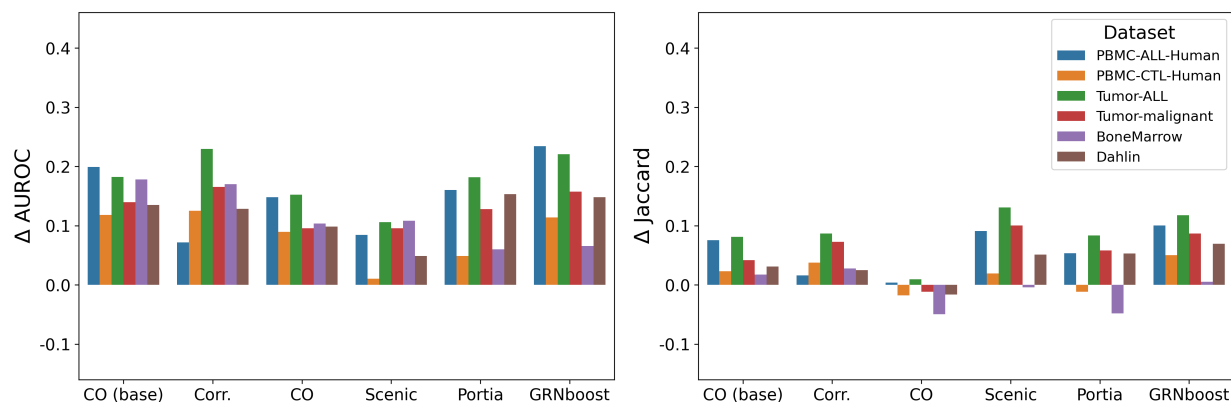


Fig. 3: Change in performance from baseline methods to GRNITE on GrouNdGAN data. Change in the area under the receiver operating characteristic curve ($\Delta AUROC$, left) and change in the Jaccard similarity of edge sets ($\Delta Jaccard$) after our method has been applied to the target graph.

these data, SCENIC experiences a large drop in precision after application of GRNITE, driven by a very low number of edge predictions in the original SCENIC method (<300 for all datasets, Supp. Fig. 18). Finally, analogous to the experiments on BEELINE, ablations with respect to using a purely biological prior ($\mathbf{A}_{bio}$, Supp. Fig. 19, 20) or purely text embedding-based prior ($\mathbf{A}_{emb}$, Supp. Fig. 21, 22) show better performance from the combined prior across the majority of metrics.

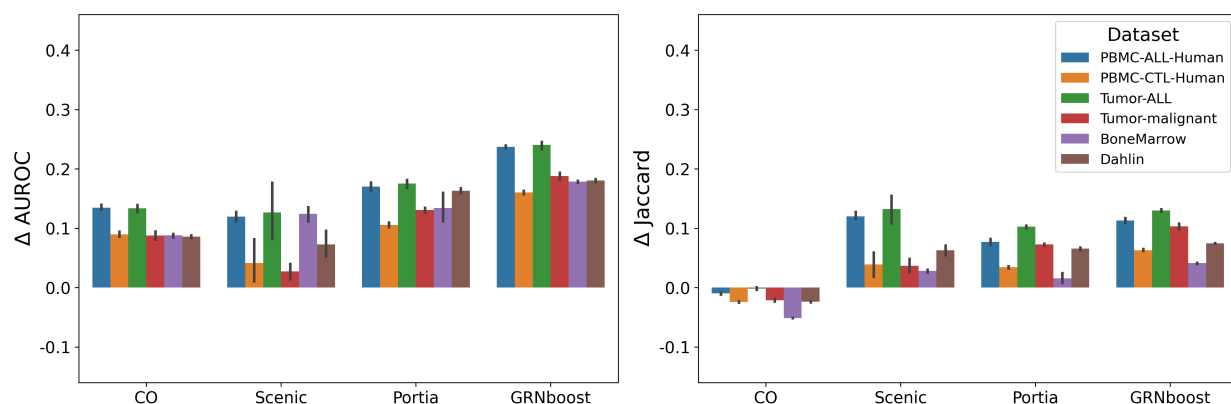


Fig. 4: Change in performance from baseline methods to GRNITE on the subsamples of GrouNdGAN data. Change in the area under the receiver operating characteristic curve ($\Delta AUROC$, left) and change in the Jaccard similarity of edge sets ($\Delta Jaccard$) after our method has been applied to the target graph. Error bars indicate variance between 5 subsample runs.

Furthermore, we produced five random subsamples, each containing 2,000 cells from the GrouNdGAN dataset, to assess the variance in the performance of GRNITE on subsets of the data. We note that the variance in the performance for all methods is small, with the exception of SCENIC, for which we observe larger variance between the AUROC on different cell subsets (Fig. 4). We omitted CellOracle base-graph and correlation network for these analyses, as these are not commonly employed inference methods, and are mainly relevant as baselines.

3.3 Increasing inter-method consensus

Next, we evaluated the fraction of shared and unique edges predicted by each of the methods on the BEELINE and GRowNdGAN datasets. We measured the Jaccard similarity between all pairs of baseline methods, as well as the pairs of final GRNs that have been processed with GRNITE. We observe a large increase in pairwise Jaccard similarity of the edge sets after applying GRNITE (Fig. 5) for all datasets and methods evaluated.

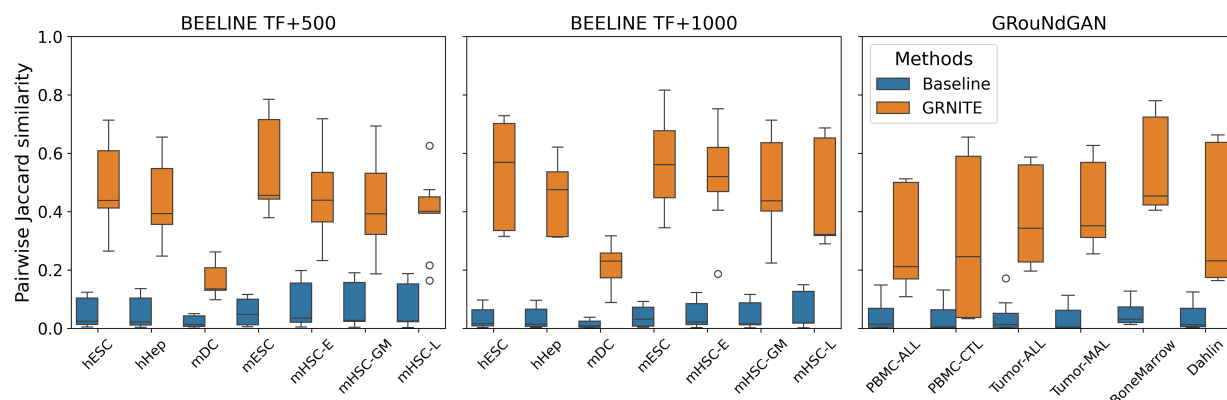


Fig. 5: Pairwise Jaccard similarities between edge sets of GRNs inferred by the baseline methods (blue) and baseline + GRNITE (orange). Box plots show variation between all pairs of methods considered.

Furthermore, we note that the resulting edge set intersections between the pairs of methods have better AUROCs with respect to the ground truth (Fig. 6).

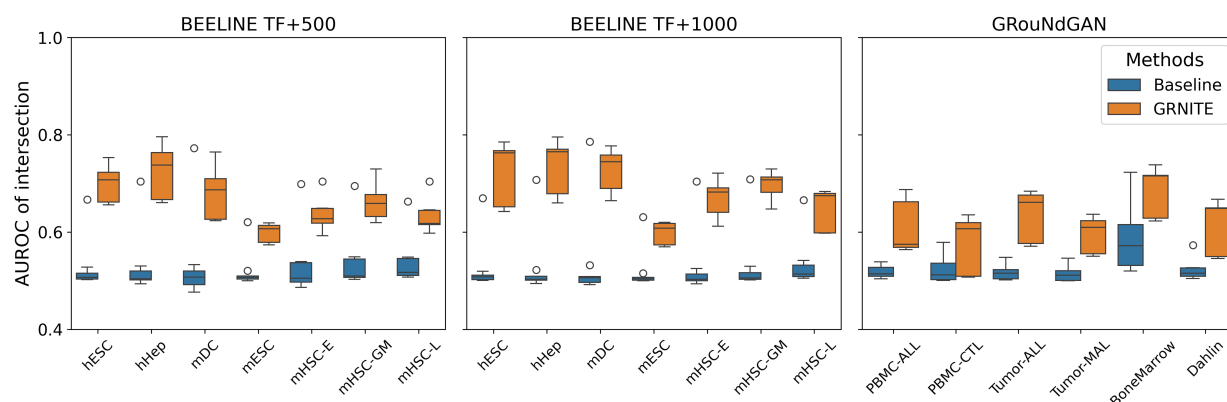


Fig. 6: AUROC of the pairwise edge set intersections for GRNs inferred by the baseline methods (blue) and baseline + GRNITE (orange). Box plots show variation between all pairs of methods considered.

These observations indicate that applying GRNITE increases inter-method consensus, and the resulting consensus edge sets show an improved AUROC.

3.4 Evaluation of GRNITE on biological single-cell datasets

To evaluate GRNITE's ability to enhance existing gene regulatory networks (GRNs) in recovering biologically meaningful regulatory interactions, we conducted a comparative evaluation across two biological single-cell

datasets (HFD and PBMC, Section 2.4). For each dataset, GRNs were inferred using four baseline methods (GRNBoost, SCENIC, Portia, and CellOracle), followed by the generation of corresponding GRNITE-enhanced networks. We quantified performance using recall, defined as the fraction of ground-truth TF-target interactions recovered by each method (see Section 2.4 for details on curated ground-truth datasets). Across both the HFD and PBMC datasets, GRNITE consistently achieved substantial improvements in recall relative to the baselines (Fig. 7).

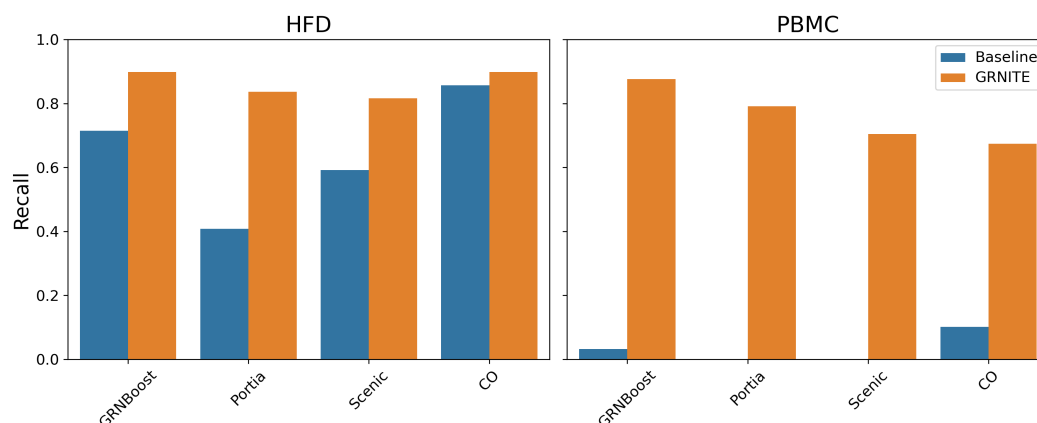


Fig. 7: Change in recall of literature and experiment supported TF-target edges for baseline methods across the HFD (left) and PBMC (right) biological single-cell datasets.

To further characterize method-specific performance, we inferred GRNs for all baseline and GRNITE-enhanced models on the full 10x Genomics PBMC dataset and evaluated changes in AUROC and Jaccard similarity (Fig. 23). Notably, Portia and SCENIC exhibited reduced performance (lower precision/recall and recall) when evaluated against the PBMC GRouNdGAN ground truth (Fig. 24). These observations are consistent with the lower recall values observed for both methods in Fig. 7.

3.5 Computational Efficiency

We evaluate the total runtime required to infer GRNs using the original base methods and compare it with the time required to run GRNITE. GRNITE consists of three steps. The first two steps construct the text- and biology-informed prior graph (Section 2.1 and 2.2), which is a one-time preprocessing step and only needs to be computed once per dataset. In the third step (Section 2.3), this prior is combined with the output of any GRN inference method to refine and enhance its predictions. Since the prior is reusable, Step 3 can be applied efficiently to different GRN inference methods without recomputing Steps 1 and 2. Runtime results in seconds are reported in Table 2 for the BEELINE TF500 and TF1000 datasets. The reported times for Step 3 are averaged over all base methods.

4 Discussion

In this study, we introduced GRNITE, a flexible meta-method designed to enhance the performance of any scRNA-seq-based GRN inference algorithm, with a particular emphasis on improving recall. Using both BEELINE and GRouNdGAN benchmarking datasets, we demonstrated that GRNITE substantially increases the recall of existing GRN predictions. Furthermore, for some methods GRNITE also improves precision, while for more conservative methods it slightly reduces precision while greatly expanding the number of correctly identified TF-target interactions. Additionally, we showed that GRNITE strengthens inter-method consensus and yields higher AUROC values when evaluated on consensus edge sets, highlighting its ability to

Table 2: Runtime (in seconds) on BEELINE datasets for TF500 and TF1000. Each dataset reports number of genes, number of cells, baseline runtime of each method, Steps 1&2 prior construction time, and GRNITE refinement time in Step 3.

(a) TF500							
	hESC	hHep	mDC	mESC	mHSC-E	mHSC-GM	mHSC-L
# Genes	910	948	821	1120	704	632	560
# Cells	758	425	383	421	1071	889	847
Corr.	1.4	1.0	0.7	1.4	1.2	0.8	0.6
CO	82.5	96.7	100.9	189.0	78.0	43.7	33.4
SCENIC	105.8	113.6	98.7	132.5	75.6	72.1	63.7
PORTIA	2.4	2.0	1.6	2.1	2.6	2.0	1.7
GRNBoost	12.0	9.3	7.8	11.7	11.4	8.1	6.7
DAZZLE	24.0	14.8	12.6	15.3	32.0	26.3	25.7
Steps 1&2	258.9	271.9	325.4	361.7	319.5	315.6	310.22
Step 3	44.20 $\pm$ 1.69	46.98 $\pm$ 1.41	39.61 $\pm$ 1.19	52.20 $\pm$ 0.61	37.56 $\pm$ 2.83	38.89 $\pm$ 2.22	38.03 $\pm$ 2.09

(b) TF1000							
	hESC	hHep	mDC	mESC	mHSC-E	mHSC-GM	mHSC-L
# Genes	1410	1448	1321	1620	1204	1132	692
# Cells	758	425	383	421	1071	889	847
Corr.	3.4	2.4	1.9	3.1	3.3	2.5	0.9
CO	139.9	149.7	156.1	265.9	130.7	77.2	39.3
SCENIC	147.2	160.8	143.0	163.5	108.3	106.4	73.5
PORTIA	3.8	3.1	2.7	3.2	3.9	3.9	2.1
GRNBoost	16.5	13.0	11.8	18.3	21.6	14.5	8.2
DAZZLE	26.4	16.0	13.9	16.4	35.0	28.9	26.6
Steps 1&2	320.3	330.35	382.1	424.5	381.9	363.2	318.5
Step 3	49.39 $\pm$ 3.06	50.73 $\pm$ 3.17	43.22 $\pm$ 1.55	62.78 $\pm$ 3.08	41.12 $\pm$ 3.54	41.40 $\pm$ 4.72	41.28 $\pm$ 1.45

reconcile and refine heterogeneous GRN outputs. This feature is particularly appealing for biological data, where analyses performed by different methods often yield vastly different inference results.

Our analyses on biological single-cell datasets illustrate that GRNITE not only boosts recall and inter-method consensus, but also enhances the ability of baseline methods to recover biologically meaningful TF–target interactions. We also indicate GRNITE’s ability to improve overall performance, characterized by improved AUROC and Jaccard similarity, of existing GRN inference methods over the biological PBMC dataset.

We observe that the time required to run GRNITE (particularly Step 3) is comparable to the cost of running the baseline methods evaluated in this study. This implies that processing of the inferred GRN with GRNITE does not inquire a significant computational cost, which makes it an attractive extra step in GRN inference and analysis pipelines. In comparison, Steps 1 and 2 are comparatively more time-consuming than Step 3, because it involves constructing the prior graph by heuristic matching of biological interactions to the gene set of interest. However, these steps can be performed in parallel with the baseline GRN inference.

Collectively, our findings position GRNITE as a robust and generalizable enhancement layer for state-of-the-art scRNA-seq GRN inference tools. By improving recall with only modest precision trade-offs, GRNITE enables existing methods to more reliably capture the underlying regulatory architecture of new datasets, ultimately advancing efforts to reconstruct accurate GRNs from single-cell transcriptomic data.

Acknowledgments This work was in part supported by the NSF grants DMS/NIGMS-2153704, DBI-2030604, IIS-2106837, EF-2126387, and CCF-2340481, and Rice’s Ken Kennedy Institute.

References

1. Abboud, R., Ceylan, I.I., Grohe, M., Lukasiewicz, T.: The surprising power of graph neural networks with random node initialization. arXiv preprint arXiv:2010.01179 (2020)
2. Aibar, S., González-Blas, C.B., Moerman, T., Huynh-Thu, V.A., Imrichova, H., Hulselmans, G., Rambow, F., Marine, J.C., Geurts, P., Aerts, J., et al.: Scenic: single-cell regulatory network inference and clustering. *Nature methods* **14**(11), 1083–1086 (2017)
3. Azizpour, A., Balaji, A., Treangen, T.J., Segarra, S.: Graph-based self-supervised learning for repeat detection in metagenomic assembly. *Genome research* **34**(9), 1468–1476 (2024)
4. Brown, G.R., Hem, V., Katz, K.S., Ovetsky, M., Wallin, C., Ermolaeva, O., Tolstoy, I., Tatusova, T., Pruitt, K.D., Maglott, D.R., et al.: Gene: a gene-centered information resource at ncbi. *Nucleic acids research* **43**(D1), D36–D42 (2015)
5. Cao, Z.J., Gao, G.: Multi-omics single-cell data integration and regulatory inference with graph-linked embedding. *Nature Biotechnology* **40**(10), 1458–1466 (2022)
6. Chen, J., Cheong, C., Lan, L., Zhou, X., Liu, J., Lyu, A., Cheung, W.K., Zhang, L.: Deepdrim: a deep neural network to reconstruct cell-type-specific gene regulatory network using single-cell rna-seq data. *Briefings in bioinformatics* **22**(6) (2021)
7. Chen, Y., Zou, J.: Genept: a simple but effective foundation model for genes and cells built from chatgpt. *bioRxiv* pp. 2023–10 (2024)
8. Chevalley, M., Roohani, Y.H., Mehrjou, A., Leskovec, J., Schwab, P.: A large-scale benchmark for network inference from single-cell perturbation data. *Communications Biology* **8**(1), 412 (Mar 2025). <https://doi.org/10.1038/s42003-025-07764-y>, <https://doi.org/10.1038/s42003-025-07764-y>
9. Chien, E., Li, M., Aportela, A., Ding, K., Jia, S., Maji, S., Zhao, Z., Duarte, J., Fung, V., Hao, C., et al.: Opportunities and challenges of graph neural networks in electrical engineering. *Nature Reviews Electrical Engineering* **1**(8), 529–546 (2024)
10. Du, J., Zhang, S., Wu, G., Moura, J.M., Kar, S.: Topology adaptive graph convolutional networks. arXiv preprint arXiv:1710.10370 (2017)
11. Hegde, A., Nguyen, T., Cheng, J.: Machine learning methods for gene regulatory network inference. *Briefings in Bioinformatics* **26**(5), bbaf470 (09 2025). <https://doi.org/10.1093/bib/bbaf470>, <https://doi.org/10.1093/bib/bbaf470>
12. Kim, D., Tran, A., Kim, H.J., Lin, Y., Yang, J.Y.H., Yang, P.: Gene regulatory network reconstruction: harnessing the power of single-cell multi-omic data. *npj Systems Biology and Applications* **9**(1), 51 (Oct 2023). <https://doi.org/10.1038/s41540-023-00312-6>, <https://doi.org/10.1038/s41540-023-00312-6>
13. Kingma, D.P.: Adam: A method for stochastic optimization. *International Conference on Learning Representations* (2015)
14. Kipf, T.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
15. Kumar, S., Mallik, A., Khetarpal, A., Panda, B.S.: Influence maximization in social networks using graph embedding and graph neural network. *Information Sciences* **607**, 1617–1636 (2022)
16. Li, P., Li, L., Nan, J., Chen, J., Sun, J., Cao, Y.: Kegni: knowledge graph enhanced framework for gene regulatory network inference. *Genome Biology* **26**(1), 294 (2025)
17. Li, S., Hua, H., Chen, S.: Graph neural networks for single-cell omics data: a review of approaches and applications. *Briefings in Bioinformatics* **26**(2) (2025)
18. Liu, T., Fang, Z.Y., Zhang, Z., Yu, Y., Li, M., Yin, M.Z.: A comprehensive overview of graph neural network-based approaches to clustering for spatial transcriptomics. *Computational and Structural Biotechnology Journal* **23**, 106–128 (2024)
19. Loers, J.U., Vermeirssen, V.: A single-cell multimodal view on gene regulatory network inference from transcriptomics and chromatin accessibility data. *Briefings in Bioinformatics* **25**(5), bbae382 (08 2024). <https://doi.org/10.1093/bib/bbae382>, <https://doi.org/10.1093/bib/bbae382>
20. Mei, S., Qin, Q., Wu, Q., Sun, H., Zheng, R., Zang, C., Zhu, M., Wu, J., Shi, X., Taing, L., et al.: Cistrome data browser: a data portal for chip-seq and chromatin accessibility data in human and mouse. *Nucleic acids research* p. gkw983 (2016)
21. Badia-i Mompel, P., Wessels, L., Müller-Dott, S., Trimbou, R., Ramirez Flores, R.O., Argelaguet, R., Saez-Rodriguez, J.: Gene regulatory network inference in the era of single-cell multi-omics. *Nature Reviews Genetics* **24**(11), 739–754 (2023)
22. Nourisa, J., Passemiers, A., Stock, M., Zeller-Plumhoff, B., Cannoodt, R., Arnold, C., Tong, A., Hartford, J., Scialdone, A., Moreau, Y., Li, Y., Luecken, M.D.: genenrib: a living benchmark for gene regulatory network inference. *bioRxiv* (2025). <https://doi.org/10.1101/2025.02.25.640181>, <https://www.biorxiv.org/content/early/2025/03/01/2025.02.25.640181>
23. Pratapa, A., Jaliha, A.P., Law, J.N., Bharadwaj, A., Murali, T.: Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nature methods* **17**(2), 147–154 (2020)

24. Qiu, J., Tang, J., Ma, H., Dong, Y., Wang, K., Tang, J.: Deepinf: Social influence prediction with deep learning. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 2110–2119 (2018)
25. Shu, H., Zhou, J., Lian, Q., Li, H., Zhao, D., Zeng, J., Ma, J.: Modeling gene regulatory networks using neural network architectures. *Nature Computational Science* **1**(7), 491–501 (2021)
26. Stock, M., Losert, C., Zambon, M., Popp, N., Lubatti, G., Hörmanseder, E., Heinig, M., Scialdone, A.: Leveraging prior knowledge to infer gene regulatory networks from single-cell rna-sequencing data. *Molecular Systems Biology* **21**(3), 214–230 (2025). <https://doi.org/https://doi.org/10.1038/s44320-025-00088-3>, <https://www.embopress.org/doi/abs/10.1038/s44320-025-00088-3>
27. Wang, J., Chen, Y., Zou, Q.: Inferring gene regulatory network from single-cell transcriptomes with graph autoencoder model. *PLoS Genetics* **19**(9), e1010942 (2023)
28. Wen, H., Ding, J., Jin, W., Wang, Y., Xie, Y., Tang, J.: Graph neural networks for multimodal single-cell data integration. In: Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining. pp. 4153–4163 (2022)
29. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* **32**(1), 4–24 (2020)
30. Yang, L., Liu, Z., Dou, Y., Ma, J., Yu, P.S.: Consisrec: Enhancing gnn for social recommendation via consistent neighbor aggregation. In: Proceedings of the 44th international ACM SIGIR conference on Research and development in information retrieval. pp. 2141–2145 (2021)
31. Yuan, Q., Duren, Z.: Inferring gene regulatory networks from single-cell multiome data using atlas-scale external data. *Nature Biotechnology* **43**(2), 247–257 (2025)
32. Yuan, Y., Bar-Joseph, Z.: Deep learning for inferring gene relationships from single-cell expression data. *Proceedings of the National Academy of Sciences* **116**(52), 27151–27158 (2019)
33. Zhang, Y., Li, M., Long, D., Zhang, X., Lin, H., Yang, B., Xie, P., Yang, A., Liu, D., Lin, J., Huang, F., Zhou, J.: Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176* (2025)
34. Zhao, Z., Verma, G., Rao, C., Swami, A., Segarra, S.: Link scheduling using graph neural networks. *IEEE Transactions on Wireless Communications* **22**(6), 3997–4012 (2023)
35. Zhao, Z., Verma, G., Swami, A., Segarra, S.: Distributed link sparsification for scalable scheduling using graph neural networks. *IEEE Transactions on Wireless Communications* (2025)
36. Zheng, R., Wan, C., Mei, S., Qin, Q., Wu, Q., Sun, H., Chen, C.H., Brown, M., Zhang, X., Meyer, C.A., et al.: Cistrome data browser: expanded datasets and new tools for gene regulatory analysis. *Nucleic acids research* **47**(D1), D729–D735 (2019)
37. Zhong, J., Li, L., Dannenfelser, R., Yao, V.: Benchmarking gene embeddings from sequence, expression, network, and text models for functional prediction tasks. *bioRxiv* pp. 2025–01 (2025)
38. Zhu, H., Slonim, D.K.: Improved gene regulatory network inference from single cell data with dropout augmentation. *PLOS Computational Biology* **21**(10), e1013603 (2025)
39. Zinati, Y., Takiddeen, A., Emad, A.: Groundgan: Grn-guided simulation of single-cell rna-seq data using causal generative adversarial networks. *Nature Communications* **15**(1), 4055 (2024)

A Supplementary figures and tables

A.1 Precision, Recall and F1 changes from baseline methods on BEELINE

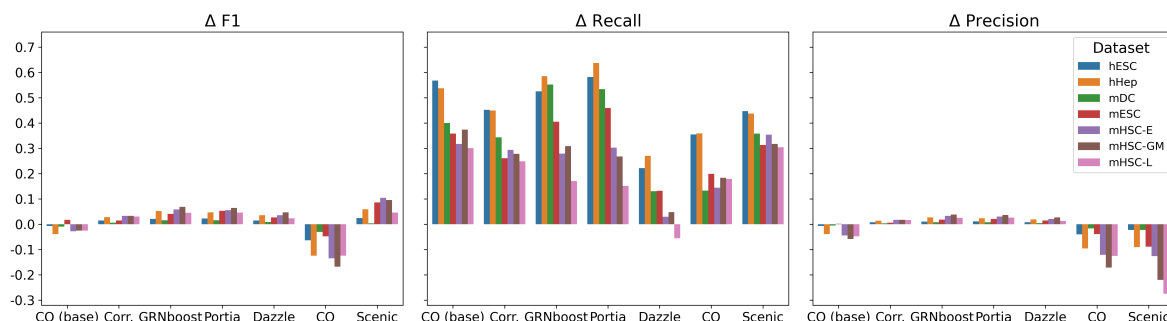


Fig. 8: Change in the F1 ($\Delta F1$, left), Recall ($\Delta Recall$, middle) and Precision ($\Delta Precision$) between the baseline GRNs and GRNITE enhanced GRNs across BEELINE TF500 datasets.

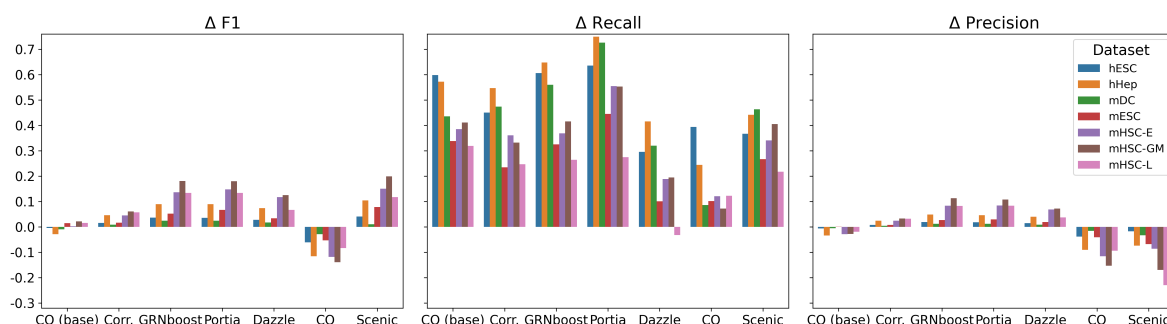


Fig. 9: Change in the F1 ($\Delta F1$, left), Recall ($\Delta Recall$, middle) and Precision ($\Delta Precision$) between the baseline GRNs and GRNITE enhanced GRNs across BEELINE TF1000 datasets.

A.2 Prior graph ablation results for BEELINE TF+500

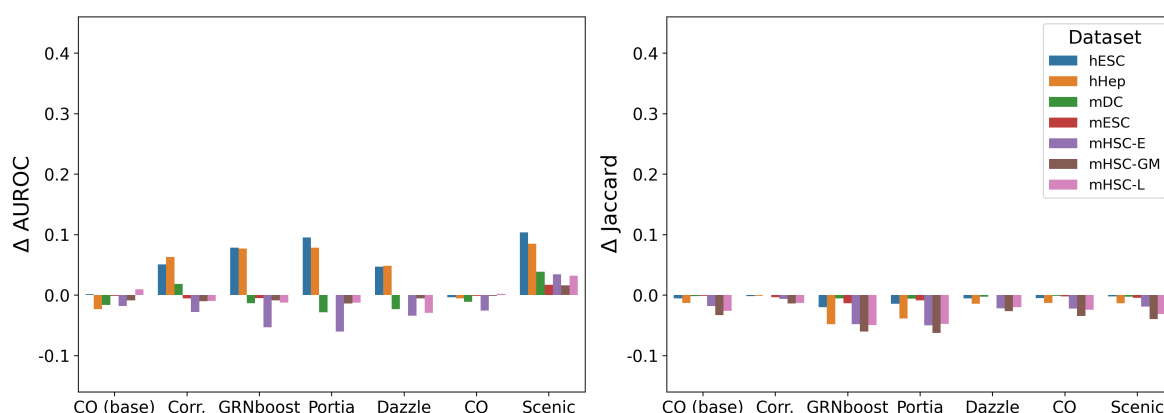


Fig. 10: Change in the area under the receiver operating characteristic curve (Δ AUROC, left) and change in the Jaccard similarity (Δ Jaccard) on using the prior graph A_{bio} for BEELINE TF500 datasets.

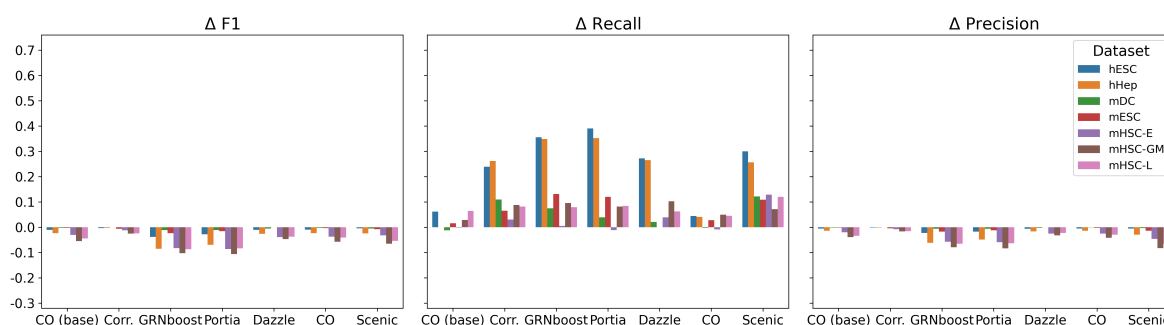


Fig. 11: Change in the the F1 (Δ F1, left), Recall (Δ Recall, middle) and Precision (Δ Precision) on using the prior graph A_{bio} for BEELINE TF500 datasets.

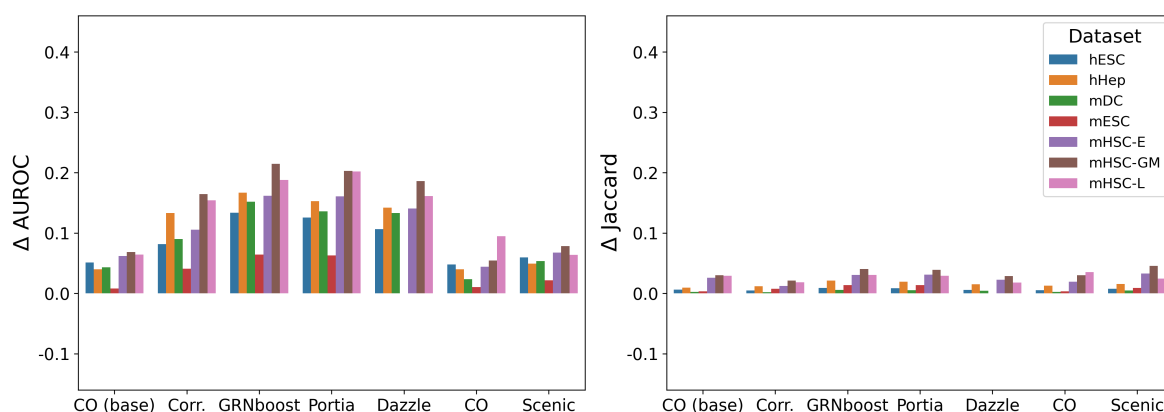


Fig. 12: Change in the area under the receiver operating characteristic curve (Δ AUROC, left) and change in the Jaccard similarity (Δ Jaccard) on using the prior graph A_{emb} for BEELINE TF500 datasets.

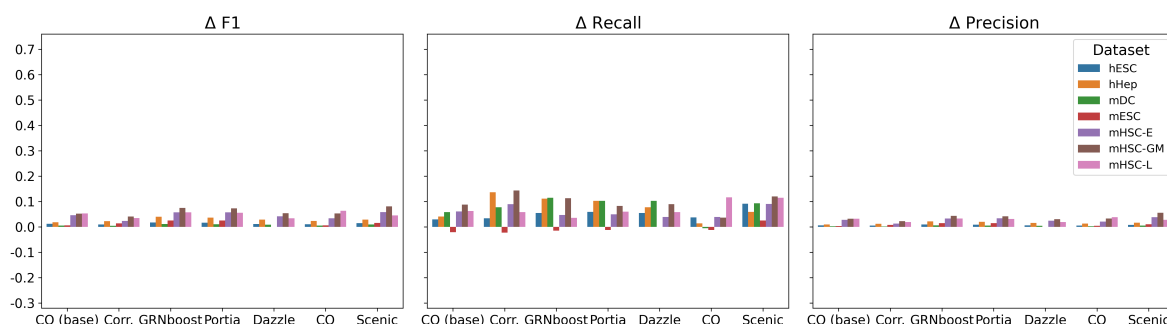


Fig. 13: Change in the the F1 ($\Delta F1$, left), Recall ($\Delta Recall$, middle) and Precision ($\Delta Precision$) on using the prior graph A_{emb} for BEELINE TF500 datasets.

A.3 Prior graph ablation results for BEELINE TF+1000

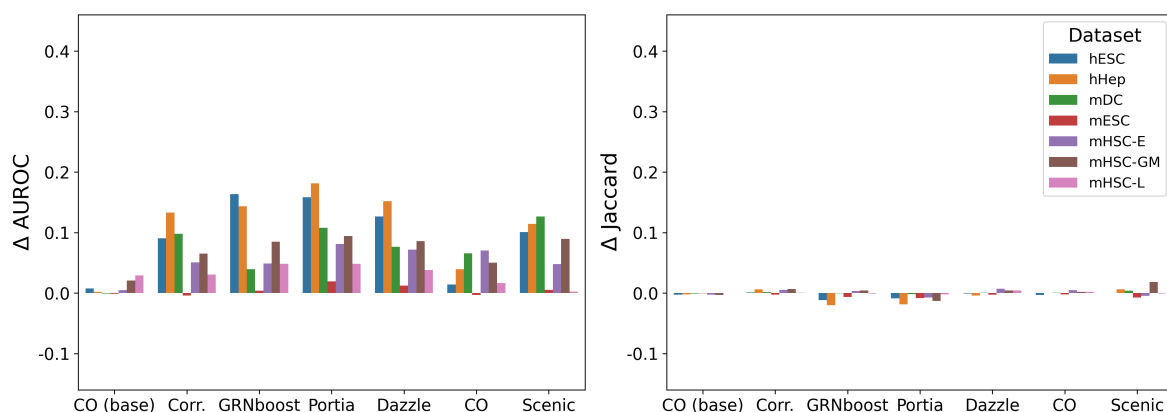


Fig. 14: Change in the area under the receiver operating characteristic curve ($\Delta AUROC$, left) and change in the Jaccard similarity ($\Delta Jaccard$) on using the prior graph A_{bio} for BEELINE TF1000 datasets.

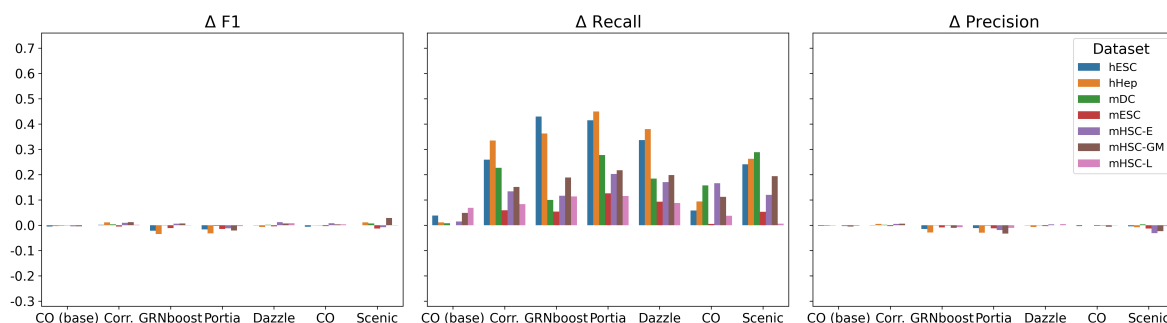


Fig. 15: Change in the the F1 ($\Delta F1$, left), Recall ($\Delta Recall$, middle) and Precision ($\Delta Precision$) on using the prior graph A_{bio} for BEELINE TF1000 datasets.

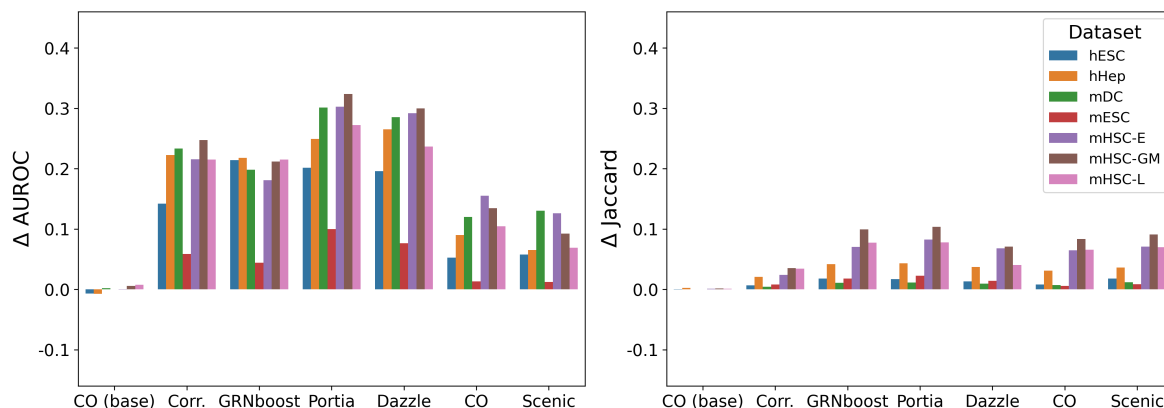


Fig. 16: Change in the area under the receiver operating characteristic curve (Δ AUROC, left) and change in the Jaccard similarity (Δ Jaccard) on using the prior graph A_{emb} for BEELINE TF1000 datasets.

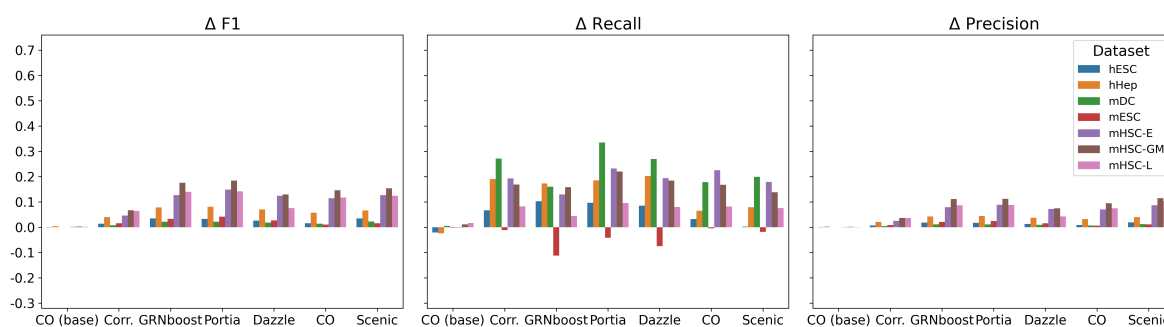


Fig. 17: Change in the the F1 (Δ F1, left), Recall (Δ Recall, middle) and Precision (Δ Precision) on using the prior graph A_{emb} for BEELINE TF1000 datasets.

A.4 Precision, Recall and F1 changes from baseline methods on GRouNdGAN

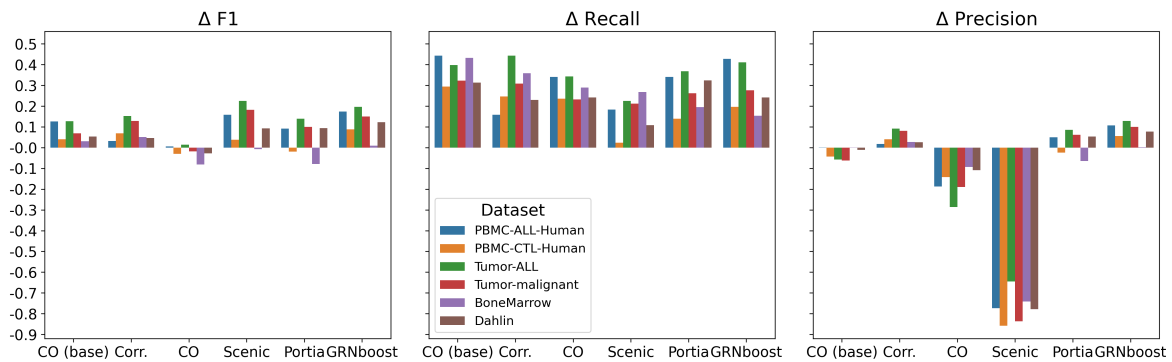


Fig.18: Change in the F1 ($\Delta F1$, left), Recall ($\Delta Recall$, middle) and Precision ($\Delta Precision$) between the baseline GRNs and GRNITE enhanced GRNs across GRouNdGAN datasets.

A.5 Prior graph ablation results for GRouNdGAN

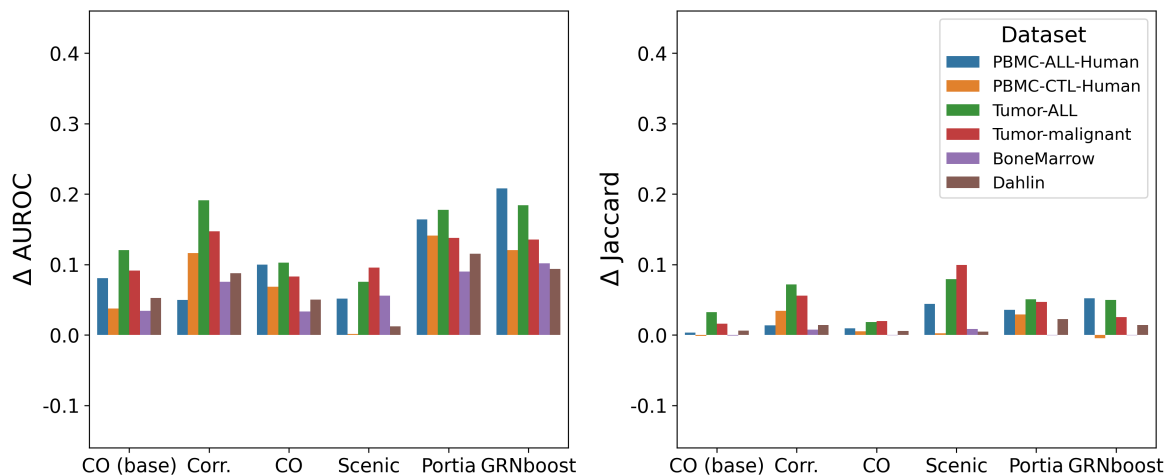


Fig.19: Change in the area under the receiver operating characteristic curve ($\Delta AUROC$, left) and change in the Jaccard similarity ($\Delta Jaccard$) on using the prior graph A_{bio} for GRouNdGAN datasets.

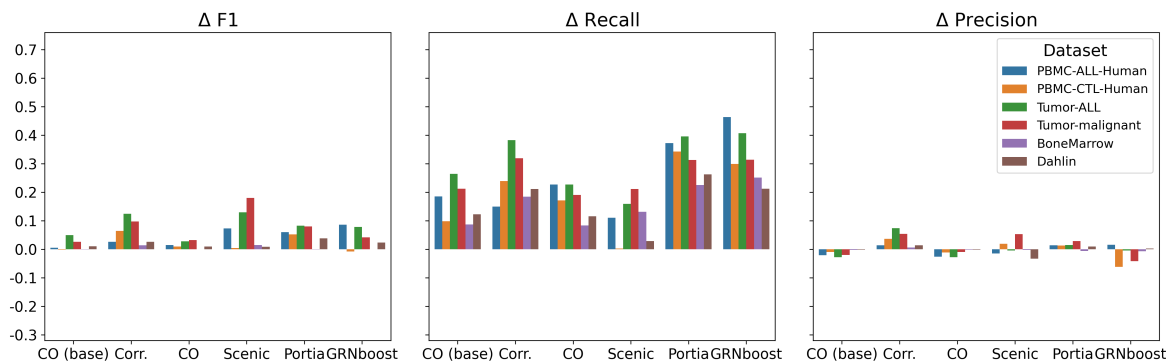


Fig. 20: Change in the the F1 ($\Delta F1$, left), Recall ($\Delta Recall$, middle) and Precision ($\Delta Precision$) on using the prior graph A_{bio} for GRouNdGAN datasets.

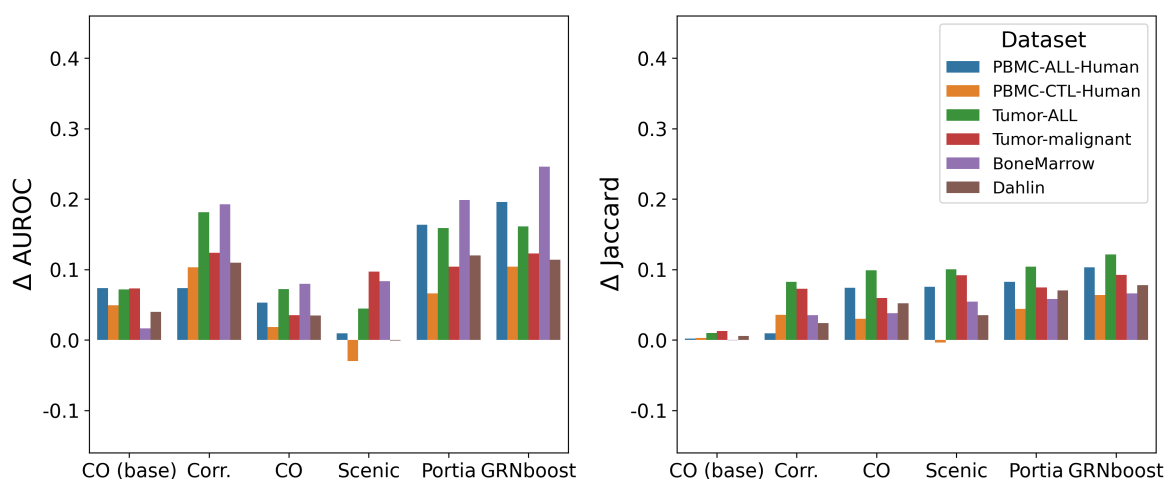


Fig. 21: Change in the area under the receiver operating characteristic curve ($\Delta AUROC$, left) and change in the Jaccard similarity ($\Delta Jaccard$) on using the prior graph A_{emb} for GRouNdGAN datasets.

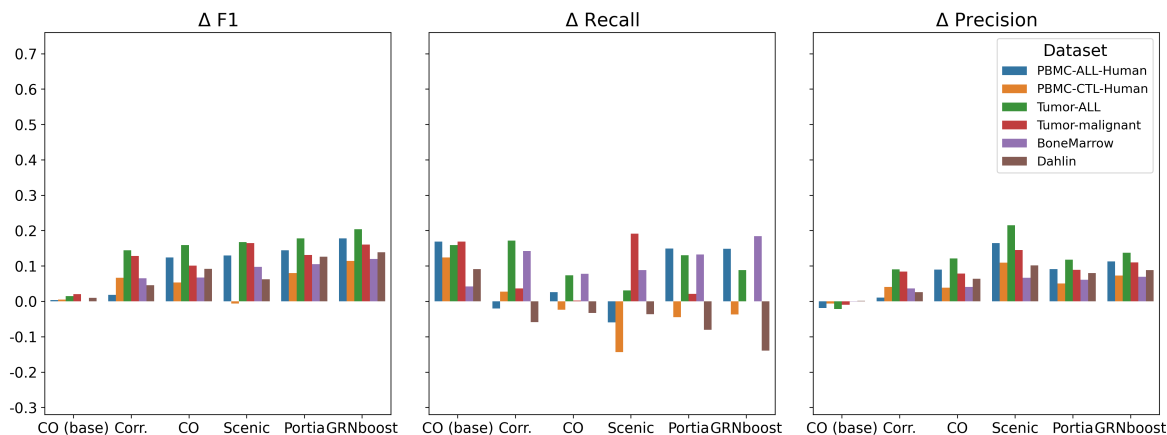


Fig. 22: Change in the the F1 ($\Delta F1$, left), Recall ($\Delta Recall$, middle) and Precision ($\Delta Precision$) on using the prior graph A_{emb} for GrouNdGAN datasets.

A.6 AUROC, Jaccard, Precision and Recall changes from baseline methods on biological PBMC

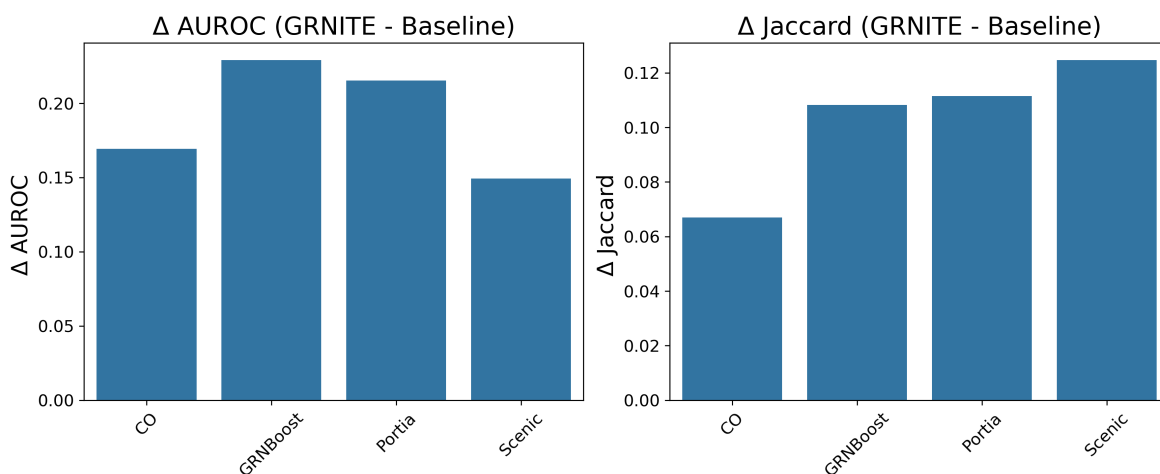


Fig. 23: Change in the area under the receiver operating characteristic curve ($\Delta AUROC$, left) and change in Jaccard similarity ($\Delta Jaccard$) between the the baseline and enhanced GRNITE GRN.

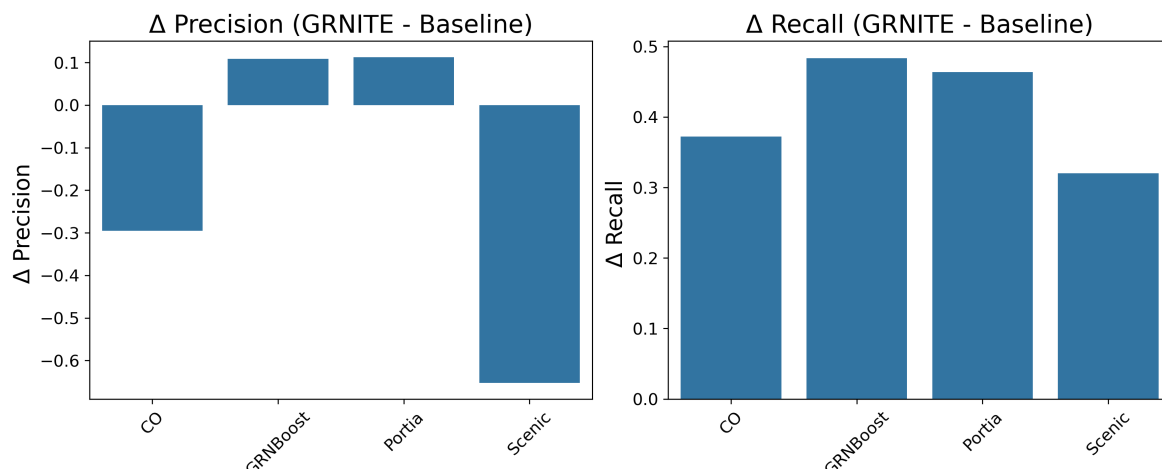


Fig. 24: Change in Precision (Δ Precision, left) and change in Recall (Δ Recall) between the the baseline and enhanced GRNITE GRN.

B Hyper-parameters

The hyperparameters used in GRNITE are kept consistent across all datasets and steps unless otherwise specified. We use the **Adam** optimizer [13] with a learning rate of $lr = 0.01$ and train the model for 5000 epochs. The graph autoencoder employs a GNN encoder with two hidden layers of dimensions [32, 32], each followed by a nonlinear **relu** activation. The feature dimensionality is reduced to $n_{low} = 100$ using TruncatedSVD prior to training for gene expressions in Step 1. During optimization, negative edge samples are generated with a ratio of **neg_multiplier**=1 relative to positive edges. The loss function in Step 2 is a weighted combination of the target and prior losses, with trade-off coefficient $\beta = 0.5$. Finally, τ_1 is set to the mean of all values in $\mathbf{W}_{emb}$ in Section 2.1 and τ_2 and τ_3 are set to 0.5 in Sections 2.2 and 2.3.

C Compute resources

All experiments were conducted on a server running Ubuntu 20.04.6 LTS, equipped with an AMD EPYC 7742 64-Core Processor and an NVIDIA A100-SXM4-80GB GPU with 80GB of memory. For model development, we utilized PyTorch version 2.5.1, along with PyTorch Geometric version 2.6.1, which also served as the source for all datasets used in our study.