Unraveling causal gene regulation from the RNA velocity graph using Velorama

Rohit Singh [*†‡] Alexander P. Wu [*†] Anish Mudide [*§] Bonnie Berger [¶‡]

March 1, 2023

Abstract

Gene regulatory network (GRN) inference that incorporates single-cell RNA-seq (scRNA-seq) differentiation trajectories or RNA velocity can reveal causal links between transcription factors and their target genes. However, current GRN inference methods require a total ordering of cells along a linear pseudotemporal axis, which is biologically inappropriate since trajectories with branches cannot be reduced to a single time axis. Such orderings are especially difficult to derive from RNA velocity studies since they characterize each cell's state transition separately. Here, we introduce Velorama, a novel conceptual approach to causal GRN inference that newly represents scRNA-seq differentiation dynamics as a partial ordering of cells and operates on the directed acyclic graph (DAG) of cells constructed from pseudotime or RNA velocity measurements. In contrast to previous approaches, Velorama is able to work directly with RNA velocity-based cell-to-cell transition probabilities. On a standard set of synthetic datasets, we first demonstrate Velorama's use with just pseudotime, finding that it improves area under the precision-recall curve (AUPRC) by 1.5–3x over state-of-the-art approaches. Using RNA velocity instead of pseudotime as the input to Velorama further improves AUPRC by an additional 1.8–3x. We also applied Velorama to study cell differentiation in pancreas, dentate gyrus, and bone marrow from real datasets and obtained intriguing evidence for the relationship between regulator interaction speeds and mechanisms of gene regulatory control during differentiation. We expect Velorama to be a critical part of the RNA velocity toolkit for investigating the causal drivers of differentiation and disease.

Software availability: `https://cb.csail.mit.edu/cb/velorama`

[*]These authors contributed equally to the work

[†]Computer Science and Artificial Intelligence Lab., MIT, Cambridge, MA 02139

[‡]Co-corresponding authors: `{rsingh,bab}@mit.edu`

[§]Phillips Exeter Academy, Exeter, NH 03883. Work performed while AM was a summer student at MIT.

[¶]Computer Science and Artificial Intelligence Lab and Dept. of Mathematics, MIT, Cambridge, MA 02139

1 Introduction

Capturing regulator–target gene interactions from expression assays, i.e., the inference of gene regulatory networks (GRNs), has been a long-standing research focus in computational biology [1–3]. In particular, single-cell RNA-seq (scRNA-seq) technologies offer enormous potential for discovering cell type-specific GRNs. We focus here on a potent innovation in scRNA-seq-based GRN inference: taking advantage of scRNA-seq datasets' ability to capture snapshots of cell states along dynamic biological processes [4]. By exploiting pseudotime computed from trajectory analysis, recent GRN inference methods have constructed an ordering of cells over which they compare the expression profile of a gene against the lagged profile of its putative regulators [5–11]. This ordering of cells with respect to time (or pseudotime) enables more accurate inference and brings us closer to discovering causal regulatory relationships rather than just gene associations.

Unfortunately, existing GRN inference methods that incorporate differentiation information have a fundamental limitation; they require a *total*, or linear, ordering over cells (or groups of cells) in order to create a "time" axis along which each gene's expression profile can be correlated with others (**Figure** 1a). For a trajectory with multiple branches, which frequently occurs with scRNA-seq, a total ordering of cells would collapse these distinct branches down into a single global sequence of cells.

A related and even more pressing challenge is incorporating RNA velocity into GRN inference methods. RNA velocity compares the spliced and unspliced counts of genes to predict per-cell state transitions during a differentiation process [12,13]. RNA velocity has proven powerful in understanding how differentiating cells commit to specific fates [14], and Qiu et al. [9] have argued that it more faithfully captures the expression dynamics of differentiation than pseudotime. These advantages suggest that the rich information encoded in RNA velocity data could be beneficial for inferring gene regulatory interactions. However, it is unclear how RNA velocity can be adapted as an input to existing GRN approaches, since they require a global pseudotime estimate (on a timeline) while RNA velocity is an inherently local measure. One workaround could entail using RNA velocity vectors to forcibly collapse cells into a global pseudotime-series over the full dataset, but doing so removes much of the rich local information provided by this data.

Here, we present Velorama, a novel method for gene regulatory network inference on scRNA-seq data that does not require a total ordering over the differentiation landscape. While some existing methods incorporate gene-wise RNA velocity for GRN inference [9, 15, 16], they do not account for how gene-wise velocities aggregate into cell state transitions. Velorama takes advantage of recent advances in cell fate mapping using RNA velocity [14] and is the first method to fully incorporate cell-to-cell transition probabilities for GRN inference. Our key conceptual advance is to perform causal GRN inference while modeling the differentiation landscape as a *partial*, rather than total, ordering of cells (**Figure** 1a). Thus, we model cell-differentiation in a local, rather than global, context, which enables us to better capture cell dynamics. Instead of a linear pseudotemporal axis, we perform inference on a directed acyclic graph (DAG) of cells that encodes the partial ordering: edges in this graph connect transcriptionally related cells, with the edge direction capturing the local direction of cell differentiation. When RNA velocity data is available, the DAG is constructed directly from the cell transition matrix, allowing us to leverage the full richness of RNA velocity data. When only pseudotime information is available, we construct the DAG by orienting the edges of a k-nearest-neighbor graph of cells in the direction of increasing pseudotime, which allows us to appropriately model merging and branching trajectories without collapsing them onto a single axis.

To address the algorithmic challenge of performing causal inference on a partial ordering, we build upon recent innovations in Granger causal (GC) inference. In a gene regulatory context, Granger causality reasons that a causal regulatory gene's past expression will be significantly predictive of its target gene's current

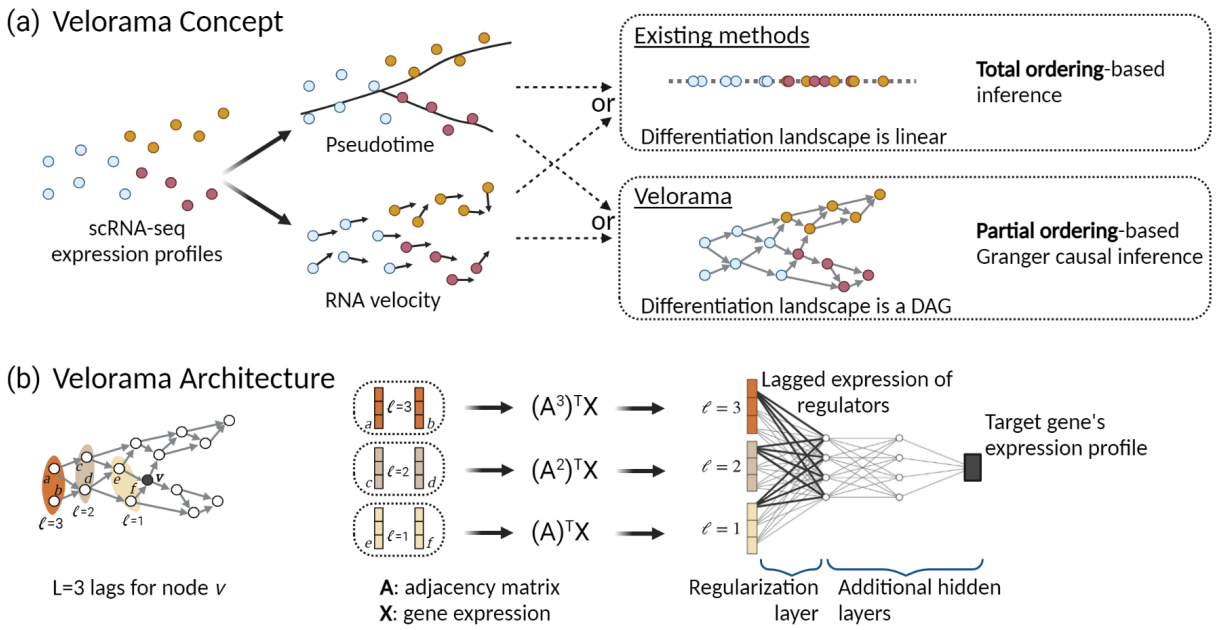

Figure 1: Schematic of Velorama: Existing methods need to impose a total (i.e., linear) ordering on cells in order to perform gene regulatory network (GRN) inference. When supplied with RNA velocity, they convert it to a global pseudotime measure and do not make use of the cell transition matrix. (**a**) Our key insight is that the differentiation landscape is more faithfully captured by a partial ordering, or directed acyclic graph (DAG), of cells constructed from RNA velocity (or pseudotime, as available) data. (**b**) We perform Granger causal inference on this DAG, taking advantage of a recent advance in generalizing Granger causality with the use of graph neural networks. At each node (cell), we collect the expression of putative regulators (e.g., three genes per cell) at the cell's ancestor nodes (e.g., we look back three lags). These are applied to predict the target gene's expression. A key component of our neural network is the first hidden layer where regularization of the weights is performed to identify statistically significant regulator–target relationships.

expression. While standard GC inference approaches are limited to total orderings, we recently introduced a generalization of Granger causality to partial orderings that uses a graph neural network framework to perform GC inference on a DAG [17]. Our previous work assessed pairwise interactions between each ATAC-seq peak (chromatin accessibility) and a candidate gene in its genomic neighborhood. However, this is insufficient for assessing gene–gene interactions, as transcriptional regulation often requires the coordination of multiple co-regulators [18]. We therefore introduce here a different mathematical formulation of GC inference that allows us to simultaneously consider *all* potential regulators and their non-linear, combinatorial effects. In Velorama, regulator–target gene comparisons are made locally for each node, by relating the target gene's value at a node to the values of *all* candidate regulators at the node's ancestors in the DAG (**Figure** 1b).

We evaluated Velorama on a variety of synthetic and real scRNA-seq datasets, on which it demonstrated significantly greater accuracy than existing methods. On synthetic datasets with associated ground-truth GRNs made available by Dibaeinia et al. [19], we first applied Velorama using only pseudotime data, benchmarking it against other GRN methods. Velorama offers state-of-the-art precision and recall, substantially outperforming a diverse set of pseudotime-based GRN inference techniques on the area under precision recall curve (AUPRC) metric. By applying Velorama on RNA velocity data, generated from the same ground-truth GRNs, we show that the use of RNA velocity can dramatically improve the quality of GRN

inference, leading to an additional 1.8-3x improvement in AUPRC over what was achieved with pseudotime. These findings suggest that Velorama can help unlock insights unattainable by current methods. When run on real datasets characterizing brain, pancreas, and bone marrow differentiation, RNA velocity-based inference captures relevant regulatory interactions comparably or better than pseudotime. As RNA velocity assays are increasingly being leveraged to reveal fine-grained dynamics of differentiation and disease, Velorama's unprecedented ability to identify causal regulatory drivers from RNA velocity data opens the door to uncovering deeper mechanistic insights into dynamic gene regulatory processes.

2 Algorithm

2.1 Background: classical Granger causal inference and its non-linear extensions

Granger causal (GC) inference, a statistical approach for estimating causal relationships within dynamical systems, has been effectively used in many settings for which only observational data is available [20–24]. Granger causality leverages the key intuition that since a cause precedes its effect, temporally-predictive relationships between dynamically changing variables could be indicative of causal relationships.

Statistical and machine learning methods have been developed to recover such temporally-predictive relationships. Broadly, there are two strategies to assess the existence of a GC relationship between variables x and y: ablation and invariance [17]. The ablation approach involves training two predictive models of y, denoted as u and $u_{\setminus x}$, where u includes the history of every variable in the system and $u_{\setminus x}$ excludes the history of x. If u performs significantly better than $u_{\setminus x}$, as determined by a one-tailed F-test, then x Granger-causes y. The invariance approach involves training just one predictive model of y, denoted as f, using the history of every variable. In this approach, x does *not* Granger-cause y if and only if the learned weights governing the interaction between x and y are all equal to 0. Equivalently, no causal relationship exists exactly when the prediction of y is invariant to the history of x. Unlike our previous work [17], here we chose the invarance-based approach since it allows multiple candidate Granger causal variables to be evaluated simultaneously, enables the lag associated with each Granger causal interaction to be determined, and reduces training time.

While the classic formulation of GC inference assumes linear interactions, extensions have been developed to capture the nonlinear interactions more often seen in real-world datasets. Many of these extensions have been based on deep learning-based architectures. Tank et al. [25] introduce a regularized multilayer perceptron and a long short-term memory network that model nonlinear relationships while simultaneously determining the lag of each putative causal relationship. Marcinkevičs and Vogt [26] extend self-explaining neural networks to multivariate time series data in order to determine whether a causal relationship induces a positive or negative effect.

Given a dynamical system with a totally-ordered sequence of N observations, each over G variables, Granger causal inference involves training per-variable models $f_1, f_2, \ldots, f_G$. Here, f_j models variable j as a function of the previous L observations:

$$z_j(t) = f_j\big(z_1(t-L;t-1), z_2(t-L;t-1), \ldots, z_G(t-L;t-1)\big) + e_j(t)$$

Here, $z_j(t)$ denotes the value of the variable j at observation t, $e_j(t)$ denotes an error term, and $z_k(t-L;t-1)$ is the sequence of L past observations of variable k: $(z_k(t-L), \ldots, z_k(t-1))$ [25]. Each pair of variables (i,j) has an associated weight tensor $\mathbf{W}_{(i,j)}$ that defines how variable j depends on past lags of variable i. The sub-tensor $\mathbf{W}^l_{(i,j)}$ refers to the lag-l value of interaction between $z_i(t-l)$ and $z_j(t)$. Variable i is then said to Granger-cause j if $|\mathbf{W}_{(i,j)}| \neq 0$, meaning that f_j is not invariant to z_i. During training, regularization

can be applied to each weight tensor $\mathbf{W}_{(i,j)}$ to assist in achieving exact zeros for the non-causal relationships. By applying additional regularization terms on each $\mathbf{W}^l_{(i,j)}$, we can automatically detect the relevant lags of each putative causal relationship [25].

2.2 Velorama: non-linear Granger causal inference on partially ordered observations

In both the linear and non-linear formulations above, z_t is defined to temporally precede z_{t+1}, which implies the requirement of a total ordering on the N observations. Thus, dynamical systems that consist of branching points, such as cellular differentiation trajectories, are not compatible with these approaches. To address this, Velorama extends the nonlinearity and automatic lag selection of modern GC methods to DAG-structured dynamical systems that encode a partial ordering of observations.

Velorama takes in two inputs, $\mathbf{A} \in \mathbb{R}^{N\times N}$ and $\mathbf{X} \in \mathbb{R}^{N\times G}$, and produces one output $\mathbf{GC} \in \mathbb{R}^{G\times G}$. $\mathbf{A}$ is the adjacency matrix of the DAG, where each node represents an observation and edges connect these observations. $\mathbf{X}$ is the feature matrix which describes the values of the G variables over the N observations, and $\mathbf{GC}$ summarizes the inferred causal graph, where $\mathbf{GC}_{ij}$ represents the strength of the causal relationship between variables i and j. We pre-compute a modified matrix $\mathbf{A}'$, defined to be the normalized transpose of $\mathbf{A}$, where row sums to 1 so as to account for variability in the in-degrees of the DAG. If a row of $\mathbf{A}^T$ consists of all zeros, the row is unaltered. $\mathbf{A}'$ serves as a graph diffusion operator that aggregates information over each observation's ancestors.

To infer Granger causal relationships, we train G separate models $f_1, f_2, \ldots, f_G$, each of which has the same architecture. We propose f_j to be a multilayer neural network that models variable j as a nonlinear function of ancestors within the DAG. The key component of our model architecture is the first hidden layer, which takes on the form

$$\mathbf{h}^{(1)} = \sigma\Big(\sum_{l=1}^{L}(\mathbf{A}')^l\mathbf{X}\mathbf{W}^{1,l} + \mathbf{b}_1\Big)$$

Here, $(\mathbf{A}')^l$ represents the lth power of $\mathbf{A}'$, $\mathbf{W}^{1,l} \in \mathbb{R}^{G\times d}$ is a learned weight matrix and $\mathbf{b}_1 \in \mathbb{R}^{N\times d}$ is a learned bias term. d is the number of hidden units per layer. $(\mathbf{A}')^l\mathbf{X}\mathbf{W}^{1,l}$ represents the information aggregated over l hops backward in the DAG (**Figure** A.1a). Note that, as the $\mathbf{A}'$ and $\mathbf{X}$ matrices are fixed, we can pre-compute $(\mathbf{A}')^l\mathbf{X} = \mathbf{A}'\big((\mathbf{A}')^{l-1}\mathbf{X}\big)$ inductively for $1 \leq l \leq L$. $\sigma(\cdot)$ is a nonlinear activation function. In a K layer model, the hidden layers $\mathbf{h}^{(k)}$ for $2 \leq k < K$ are given by

$$\mathbf{h}^{(k)} = \sigma\big(\mathbf{h}^{(k-1)}\mathbf{W}^k + \mathbf{b}_k\big)$$

where $\mathbf{h}^{(k)} \in \mathbb{R}^{N\times d}$, $\mathbf{W}^k \in \mathbb{R}^{d\times d}$ and $\mathbf{b}_k \in \mathbb{R}^{N\times d}$ (**Figure** A.1b). Finally, the output $f_j \in \mathbb{R}^N$ of the autoregressive model is given by

$$s\big(\mathbf{h}^{(K-1)}\mathbf{W}^K + \mathbf{b}_K\big)$$

where $\mathbf{W}^K \in \mathbb{R}^{d\times 1}$ and $\mathbf{b}_K \in \mathbb{R}^{N\times 1}$. $s(\cdot)$ is an optional element-wise decoder function that links the real numbers to a domain-specific output.

Granger causal inference via regularization of $\mathbf{W}^1$. To infer causality with respect to variable j, we concatenate the per-lag matrices $\mathbf{W}^{1,l}$ from model f_j into $\mathbf{W}^1 \in \mathbb{R}^{L\times G\times d}$. Let $\mathbf{W}^1_i \in \mathbb{R}^{L\times d}$ denote the weights that govern the interaction between variable i (regulator) and variable j (target) (**Figure** A.1c). Variable i does *not* Granger-cause variable j if $\big\|\mathbf{W}^1_i\big\|_F = 0$, where $\|\cdot\|_F$ denotes the Frobenius matrix

norm. Similarly, we can infer these causal interactions with respect to a specific lag point as well, thereby enabling automatic lag selection in our framework. Let $\mathbf{W}_i^{1,l} \in \mathbb{R}^d$ denote the weights that govern the interaction between variable i and variable j at a particular lag l. If variable i Granger-causes j, then l is not a relevant lag when $\left\|\mathbf{W}_i^{1,l}\right\|_2 = 0$, where $\|\cdot\|_2$ denotes the ℓ_2 norm.

To induce a sparse set of causal interactions, we simultaneously reduce the prediction error of each model f_j while encouraging the presence of exact zeros in the $\mathbf{W}^1$ matrices. Therefore, we define our loss function $\mathcal{L}$ to be the sum of the mean-squared error loss $\ell(\cdot,\cdot)$ over all j models and a regularization penalty applied to the $\mathbf{W}^1$ matrices over all models:

$$\mathcal{L} = \sum_{j=1}^{G} \ell(f_j, \mathbf{X}_{:j}) + \lambda R(\mathbf{W}).$$

Here, $\mathbf{X}_{:j} \in \mathbb{R}^N$ represents the value of variable j over all N observations. Let $\mathbf{W}^{(j)}$ denote the $\mathbf{W}^1$ corresponding to model f_j and $\mathbf{W} = [\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(G)}] \in \mathbb{R}^{G\times L\times G\times d}$ represent the concatenation of these matrices over all G models. R: $\mathbb{R}^{G\times L\times G\times d} \to \mathbb{R}$ is the regularization loss function. The hyperparameter λ controls the regularization strength.

The specific form of R is guided by the regularization characteristics desired by the user. Tank et al. [25] introduced three regularization schemes: group, lag-selection, and hierarchical, with the last one being the broadest. Here, we use the hierarchical formulation, though our implementation makes all three available (**Appendix** A.2). Hierarchical regularization penalizes longer lags more heavily than shorter lags, which has been found to achieve robustness with respect to the choice of the maximum lag L [25] (**Figure** A.1d).

$$R(\mathbf{W}) = \sum_{l=1}^{L} \left(\sum_{i=1}^{G} \sqrt{\sum_{j=1}^{G} \left\| \left(\mathbf{W}_i^{(j),l}, \ldots, \mathbf{W}_i^{(j),L}\right) \right\|_F^2} \right)$$

Traditional gradient descent algorithms such as Adam [27] and stochastic gradient descent often fail to converge the learned weights to exact zeros, hindering causal detection. We thus optimize the objective function $\mathcal{L}$ via proximal gradient descent, a specialized algorithm designed to induce sparsity [28].

2.3 Constructing the DAG from pseudotime or RNA velocity data

We first learn low-dimensional representations for each cell's gene expression profile by performing principal component analysis (PCA) and projecting the gene expression profiles of cells onto the top M principal components. Optionally, if time stamps are available as is the case with scRNA-seq time series data, users can first endow the learned representations with this temporal information by using multimodal integration techniques like Schema [29] (**Appendix** A.5). We then construct a k-nearest-neighbor graph $\mathcal{G}$ of cells based on Euclidean distances in the representation space. When using Velorama with pseudotime, we orient each edge $e \in \mathcal{G}$ in the direction of increasing pseudotime. Doing so preserves the underlying differentiation structure while ensuring that the constructed graph is acyclic.

When using Velorama with RNA velocity, we first apply CellRank to calculate cell–cell transition probabilities [14]. The probability p_{ij} of cell i transitioning into a neighboring cell j is calculated based on the alignment between the velocity vector of cell i and the displacement vector between cell i's and j's gene expression profiles. Consequently, these transition probabilities characterize cellular dynamics in a cell-specific, local manner that is conducive for representing cell state changes on single-cell graphs. As such, we directly use these transition probabilities as edge weights for the Velorama adjacency matrix $\mathbf{A}$.

A possible complication with directly using the CellRank cell–cell transition matrix in Velorama is that this matrix may contain cycles. In Velorama, we only consider ancestors up to L hops away, so only cycles of up to length L are relevant. To combat the possibility of cycles in these L hops, we first remove cycles of length 2 by setting $p_{ji} = 0$ if $p_{ij} > p_{ji}$. When we pre-compute powers of $\mathbf{A}'$, we then set all non-zero diagonal terms back to zero after each successive multiplication with $\mathbf{A}'$, as cycles can be identified by non-zero terms along the diagonal.

2.4 Identification of Granger-causal interactions

The hyperparameter λ plays a key role in regularization, with higher values of λ leading to sparser GRNs. However, selecting the appropriate hyperparameters for a specific setting can be a challenge. To address this, we sweep over a range of λ values and compute the GRN by aggregating over their results. Thus, the user can simply use Velorama with its default settings. We sweep λ uniformly over the log-scaled range of $[0.01, 10]$. To aggregate results over the various λ, we retain only those $\mathbf{GC}$ matrices (recall $\mathbf{GC} \in \mathbb{R}^{G \times G}$) with between 1% and 95% non-zero values. We then sum the retained $\mathbf{GC}$ matrices corresponding to each lag, which results in L composite $\mathbf{GC}$ matrices (one for each lag). These per-lag matrices are themselves summed to produce a single $\mathbf{GC}$ matrix, which we report as the GRN.

Training details. We trained using proximal gradient descent, which minimizes the sum of the mean squared error and the hierarchical regularization. We used ReLU [30] for the activation function $\sigma(\cdot)$ and the identity function for the link function $s(\cdot)$. For all results, we used a single hidden layer with 16 nodes, a learning rate of 0.01, and a maximum of 10,000 epochs with early stopping as implemented by Tank et al. [25]. The hyperparameter sweep of λ covered 19 log-spaced values in the range $[0.01, 10]$. Model training was performed on a mix of Nvidia V100 (32 GB memory) and A100 (80 GB memory) GPUs.

Scalability and runtime. We perform $G\Lambda$ training runs where G is the number of target genes and Λ is the number of λ choices we sweep over. Each run is relatively light and we were able to perform 10 parallel runs for λ settings on a single V100 GPU. While N (the number of cells) is not a major runtime scalability concern, the input matrices (of size $O(NG')$, where G' is the number of candidate regulators) need to fit into GPU memory. For large datasets (e.g., with over 100,000 cells), we recommend sketching the data [31, 32] to first extract a smaller representative sample. When the list of potential targets or regulators is known, the scope of computation can be constrained. For example, in our analysis of biological data, we limited candidate regulators to the set of transcription factors in the dataset. The $(\mathbf{A}')^l\mathbf{X}$ inputs to the first hidden layer are pre-computed and amortized over all target genes, and thus this computation is not meaningful to the overall runtime. For a synthetic dataset with 1,800 cells and 100 genes, a full run of Velorama sweeping across all λ values required approximately 3 minutes on an Nvidia V100 GPU. For real scRNA-seq datasets with between 2,531–5,780 cells, 245–499 candidate regulators, and 1,585–1,873 candidate target genes, each full run took roughly 1–3 hours on a Nvidia A100 GPU.

3 Results

Datasets. We benchmarked Velorama on a series of synthetic datasets for which the underlying ground-truth GRN is known. We used four differentiation datasets from SERGIO [19], which simulates single-cell expression dynamics according to a given GRN. These datasets, which we obtained from the SERGIO GitHub repository, span a range of differentiation landscapes, including linear (Dataset A), bifurcation

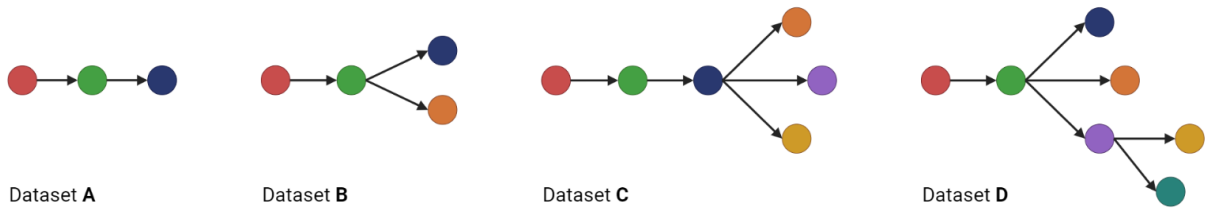

Figure 2: Ground-truth differentiation trajectories for the four synthetic datasets sourced from SERGIO. Note that the differentiation landscapes get increasingly more complex.

(Dataset B), trifurcation (Dataset C), and tree-shaped (Dataset D) trajectories (**Figure** 2, **Appendix** A.3.1). The underlying GRN for each of these datasets consists of 100 genes and 137 edges. For each of these GRNs, SERGIO simulates spliced and unspliced counts as well as total transcript counts. We also applied Velorama on three RNA velocity studies characterizing mouse endocrinogenesis, mouse dentate gyrus neurogenesis, and human hematopoiesis [33–35]. We used processed *Scanpy* [36] objects for these datasets, which were made available by scVelo and CellRank [14, 37]. In addition, we sought to evaluate Velorama's ability to incorporate experimental time-point information when available, and applied it on two time-series scRNA-seq assays profiling cardiac hypertrophy progression [38] and cancer cell line drug responses [39], respectively; both datasets only provided pseudotime (not RNA velocity) information. For all datasets, we constructed the Velorama DAGs using $k = 15$ nearest neighbors and $M = 50$ principal components; Velorama's effectiveness is robust to these hyperparameter choices (**Appendix** A.6).

Velorama accurately infers GRNs from expression dynamics. We first sought to evaluate Velorama when only pseudotime data is available. Any benchmarking of Velorama requires this setting since existing methods can not make use of the full RNA velocity transition matrices. We compared against a diverse set of methods: GENIE3, GRNBoost2, GRISLI, Scribe, PIDC, SCODE, SINCERITIES, and SINGE [5–11, 15, 16]. We chose these methods because (a) their underlying mathematical and statistical techniques vary widely (decision trees, differential equations, information theory, and total ordering-based Granger causality); and (b) they were the top-rated methods in a comprehensive single-cell GRN benchmarking study [40].

On both the area under receiver operating curve (AUROC) and the area under precision-recall curve (AUPRC) metrics, Velorama outperforms all other methods (**Table** 1). Velorama's substantial accuracy gains with respect to AUPRC are particularly notable, as the AUPRC metric has typically been the focus of GRN inference tasks due to the sparsity of GRNs and the consequent class imbalance associated with inferring its structure. On the area under receiver operating curve (AUROC) metric, Velorama also offers state-of-the-art performance, with it and GENIE3 being the two top-performing methods. GENIE3, originally designed for bulk RNA-seq, has consistently been reported to be one of the top-performing GRN methods. As such, Velorama's outperformance against GENIE3 on the AUPRC metric and competitiveness on the AUROC metric is notable. The improvement of Velorama over SINGE and SINCERITIES, both of which utilize total ordering-based Granger causal inference, also suggests that our innovation of extending Granger causality to the DAG plays a crucial role in Velorama. We note that Velorama's relative AUPRC outperformance is most substantial for Datasets C and D, which feature the most complex differentiation landscapes with the largest number of cell types and branching points in their respective trajectories (**Figure** 2). These findings support the motivating intuition behind our work: **compared to total orderings, partial orderings can more effectively capture a complex differentiation landscape**.

Method	Dataset **A**		Dataset **B**		Dataset **C**		Dataset **D**	
	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC
SINGE	0.014	0.492	0.025	0.542	0.013	0.471	0.014	0.491
GRISLI	0.015	0.440	0.013	0.468	0.014	0.488	0.013	0.443
SINCERITIES	0.017	0.568	0.018	0.555	0.012	0.454	0.016	0.591
SCODE	0.019	0.539	0.019	0.599	0.013	0.508	0.018	0.564
PIDC	0.058	0.696	0.045	0.724	0.042	0.704	0.043	0.695
Scribe	0.026	0.607	0.014	0.509	0.022	0.575	0.016	0.454
GRNBoost2	0.052	0.719	0.046	0.729	0.048	0.807	0.054	0.669
GENIE3	0.107	0.799	0.104	0.846	0.110	0.873	0.129	0.857
Velorama	**0.191**	**0.848**	**0.154**	**0.856**	**0.327**	**0.882**	**0.296**	**0.872**

Table 1: Gene regulatory network inference accuracy on SERGIO datasets. Pseudotime data is used in this comparison for methods that require an ordering of cells; see Table 2 for RNA velocity-based comparisons.

Velorama capitalizes on RNA velocity measurements. We also assessed Velorama's ability to leverage RNA velocity for GRN inference when this data is readily accessible. To do so, we applied Velorama to the same synthetic datasets as above while taking advantage of the spliced and unspliced transcript counts provided by SERGIO. We weighed edges in the Velorama DAG according to the RNA velocity-based cell–cell transition probabilities estimated by CellRank and compared the GRN inference accuracy of this RNA velocity-based version of Velorama with that of the pseudotime-based version. We also performed an ablation on RNA velocity-based inference to test if Velorama could exploit the fine-grained information from the CellRank transition probabilities, rather than a binary edge-presence indicator. We refer to the setting which directly uses the CellRank transition probabilities as the "probabilistic" RNA velocity setting. In the ablated setting, which we refer to as the "discrete" setting, we retain only the edges in the CellRank transition matrix that correspond to the top 50% of each cell's transition probabilities and binarize the resulting matrix prior to DAG construction.

Both RNA velocity settings of Velorama substantially improved upon the pseudotime version, even as the latter itself demonstrated state-of-the-art performance compared to existing benchmarks. Across the various datasets, the RNA velocity-based AUPRC metrics consistently improved upon the pseudotime-based metrics by 1.8–3x (**Table** 2). In addition, our ablation studies suggest that Velorama can exploit fine-grained cell transition probabilities derived from RNA velocity, with the probabilistic setting outperforming the discrete setting with respect to AUPRC for 3 of the 4 datasets. Indeed, with the probabilistic setting, Velorama's improvement over existing methods is substantial, surpassing AUPRC values for GENIE3 predictions by 3–5x (**Tables** 1,2). Velorama's RNA velocity-based invocation also substantially outperforms existing RNA velocity-based approaches. For instance, RNA-ODE [16] uses random forests (like GENIE3) on RNA velocity data to infer GRNs. Its performance is similar to that of GENIE3, while Velorama's AUPRC is nearly 4x higher. To confirm that these results were not a measurement artifact, we also visualized the actual ground-truth GRN matrix and the Velorama-inferred GRNs when using pseudotime or RNA velocity measurements (**Figures** 3, **Appendix** A.2). RNA velocity-based GRNs indeed capture the ground-truth regulatory relationships with higher fidelity.

The substantial performance gains associated with using RNA velocity in Velorama emphasize the value

of accurately reconstructing cell-state transitions from scRNA-seq snapshots. Our results corroborate previous findings that RNA velocity better captures temporal couplings between single-cell measurements compared to pseudotime, thereby enabling more accurate causal GRN inference [9]. For instance, GRISLI's RNA velocity-based invocation also outperforms its pseudotime-based usage. Furthermore, given single-cell time series data, Velorama can incorporate the time stamps from these datasets to better estimate temporal relationships between cells (**Appendix** A.5), driving additional improvements in GRN inference (**Appendix** A.8). Altogether, these results point to the flexibility of Velorama to make full use of the rich temporal information revealed by scRNA-seq data to unravel regulatory drivers of complex differentiation landscapes.

Method	Dataset **A**		Dataset **B**		Dataset **C**		Dataset **D**	
	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC
GRISLI (pseudotime)	0.015	0.440	0.013	0.468	0.014	0.488	0.013	0.443
GRISLI (velocity)	0.016	0.539	0.017	0.577	0.017	0.535	0.016	0.523
RNA-ODE (velocity)	0.104	0.795	0.093	0.847	0.124	0.875	0.116	0.857
Velorama (pseudotime)	0.191	0.848	0.154	0.856	0.327	0.882	0.296	0.872
Velorama (velocity, discrete)	**0.402**	0.892	0.452	**0.900**	0.572	0.908	0.455	0.894
Velorama (velocity, probabilistic)	0.399	**0.900**	**0.516**	0.856	**0.596**	**0.923**	**0.541**	**0.926**

Table 2: GRN inference using RNA velocity data on SERGIO datasets. In addition to Velorama, we also show the performance of RNA-ODE and GRISLI. The latter's pseudotime-based results are reproduced from Table 1 for convenience.

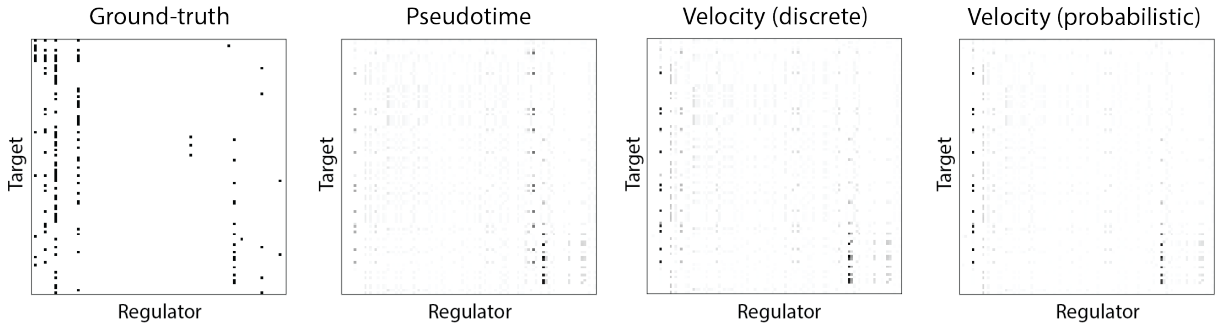

Figure 3: Ground-truth GRN for Dataset **D** and the Velorama-inferred GRNs when using pseudotime or RNA velocity. Dataset **D** includes 100 genes, and each panel is a 100x100 scatterplot showing regulator–target interactions. GRN interactions inferred using Velorama with RNA velocity cell–cell transition probabilities are qualitatively more similar to the ground-truth compared to those inferred using Velorama with pseudotime.

Velorama identifies relevant regulatory interactions in real RNA velocity differentiation datasets. We applied Velorama to analyze gene regulatory interactions related to three scRNA-seq differentiation datasets. These datasets characterize the expression dynamics for mouse endocrinogenesis [33], mouse dentate gyrus neurogenesis [34], and human hematopoiesis [35] (**Figure** 4a). After performing standard pre-processing (**Appendix** A.3.2), we applied Velorama with probabilistic RNA velocity, to infer regulator–target gene interactions for each dataset. We separately also computed a diffusion pseudotime estimate for these datasets and applied Velorama with pseudotime. For each dataset, we first identified the top 5% of all candidate regulator–target gene interactions based on the same scoring scheme as above. The regulator genes implicated in these interactions were significantly enriched for Gene Ontology (GO) terms corresponding to the

specific biological processes being profiled in each of these datasets (FDR < 0.05 Enrichr [41], **Figure** 4b).

Moreover, when comparing this enrichment for interactions inferred when pseudotime instead of RNA velocity was supplied to Velorama, the RNA velocity-based interactions yielded more significant enrichment for 5 of the 7 GO terms. This improvement in enrichment is possibly a consequence of the complex expression dynamics of these datasets, comprising multiple cell types and differentiation branches, which may be challenging for previous pseudotime-based approaches that collapse these dynamics onto a total ordering of cells. Notably, even with pseudotime, employing Velorama's partial ordering-based representation helps alleviate these issues, as evidenced by the significant enrichment results for that setting. Nonetheless, the RNA velocity-based DAG better handles these dynamics. Taken together, these results provide evidence for both Velorama's effectiveness in detecting relevant regulatory relationships as well as the benefits of leveraging RNA velocity to do so.

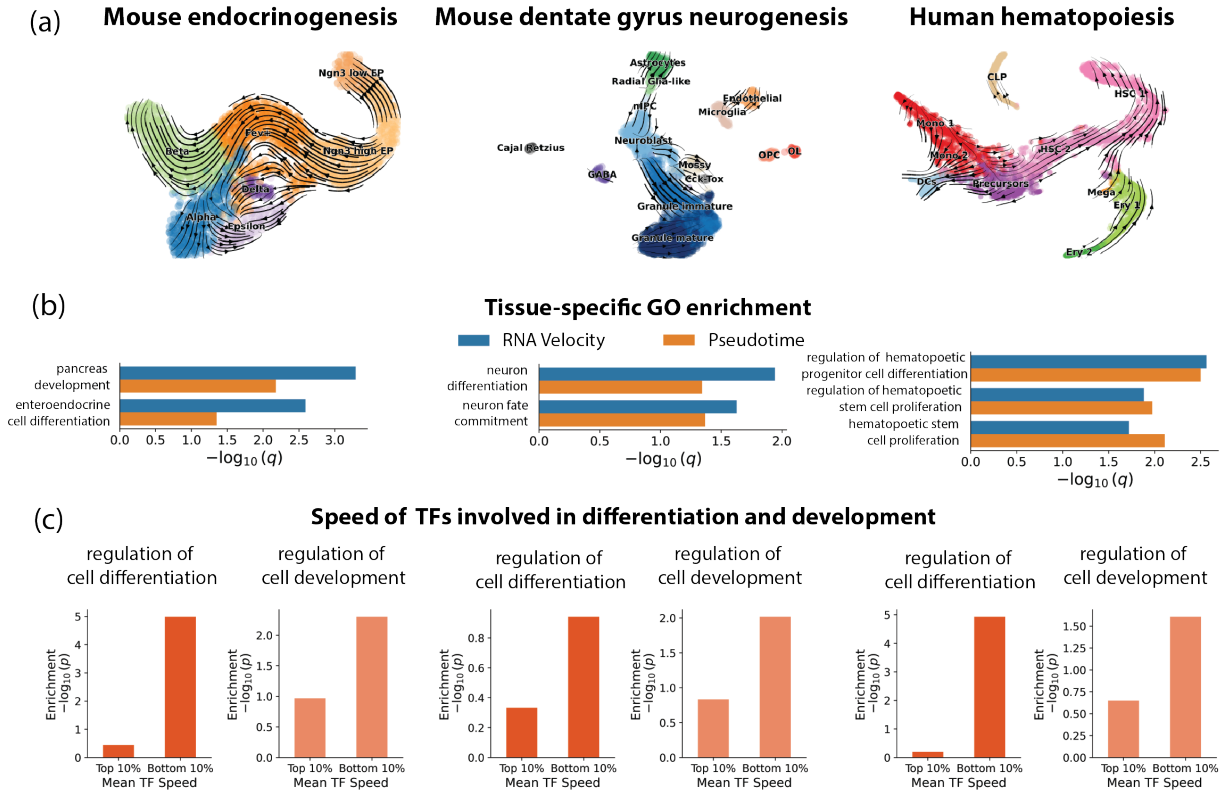

Figure 4: Velorama applications for studying regulatory dynamics on scRNA-seq RNA velocity datasets. (a) Visualizations of expression dynamics. Note that mouse endocrinogenesis has fewer branching/separated trajectories. (b) Enrichment of dataset-specific biological pathways for transcriptional regulators identified by Velorama. Compared to pseudotime-based inference, RNA velocity-based inference is more strongly enriched for regulators known to be involved in relevant pathways. (c) Number of enriched differentiation GO terms for the fastest and slowest 10% of TFs identified by Velorama. Differentiation and development-related TFs are associated with slower interaction speeds.

Velorama suggests development/differentiation regulators act more slowly than others. Velorama's ability to estimate the relevant lags for Granger causal interactions enables it to predict the speed of each regulator–target gene interaction (**Appendix** A.9). We calculated the interaction speeds for the top 5% of regulator–target gene interactions in each of the three differentiation datasets. For each of the datasets, the

bottom 10% of regulators by interaction speed were significantly enriched for GO terms related to cellular differentiation (GO:0045595) and development (GO:0060284) ($p < 0.05$, hypergeometric test, **Figure** 4c). In comparison, the regulators with the highest 10% of interaction speeds did not show enrichment for these terms. This pattern of enrichment for only the lower interaction speed regulators was observed not just for each tissue but across all three tissues.

This observation aligns with known mechanisms of gene regulatory control during differentiation. Regulators of these processes often participate in a number of epigenetic and/or cofactor recruitment steps prior to influencing their target [42, 43]. As a result, these intermediary regulatory activities may manifest in a lag between changes in regulators' expression levels and those of their targets, hence leading to lower observed interaction speeds for such regulators. Notably, we did not observe a similar phenomenon for housekeeping transcription factors. Identifying the regulatory range and duration of transcription factors is a major open problem and this preliminary investigation suggests a promising direction of future work: RNA velocity measurements, analyzed with Velorama, could systematically measure regulatory speed in a variety of tissues, leading to a deeper understanding of gene regulatory control.

Velorama flexibly incorporates time stamps from real time series datasets Many single-cell experiments provide time stamp information [44], which we hypothesized could be incorporated into Velorama to better capture temporal relationships between cells and, hence, more accurate GRNs. We analyzed two such single-cell time series datasets. The first profiled the progression of cardiac hypertrophy in cardiomyocytes [38], while the second profiled the response of A549 cells to pracinostat, an HDAC inhibitor [39]. In both cases, RNA velocity information was unavailable and we therefore applied Velorama with pseudotime to infer regulator–target gene interactions. We leveraged Schema [29] to synthesize the datasets' time stamps with gene expression data for a more accurate pseudotemporal ordering, subsequently applying Velorama on the synthesized data (see **Appendix** A.5 for details). As in our previous analyses, we identified the top 5% of all candidate regulator–target gene interactions and evaluated the retained regulator genes for enrichment of GO terms related to the specific biological processes profiled in each dataset. For the cardiac hypertrophy dataset, we evaluated genes related to the regulation of cardiac muscle hypertrophy (GO:0010611) and focused on genes implicated in chromatin organization (GO:0006325) and remodeling (GO:0006338) for the A549 dataset (given the impact of HDAC inhibitors on these processes). Even with no time stamp information, we observed a moderate enrichment for these GO terms (FDR < 0.1, Enrichr [41]). However, incorporating time stamps contributed to a stronger enrichment (**Table** 3). These results further highlight how Velorama can be flexibly used to unlock rich temporal information within single-cell data to reveal crucial regulatory interactions underlying disease and treatment-related processes.

Method	cardiac hypertrophy	A549 + pracinostat	
	regulation of cardiac muscle hypertrophy	chromatin organization	chromatin remodeling
Velorama (pseudotime)	2.2×10^{-2}	8.9×10^{-2}	7.7×10^{-2}
Velorama (pseudotime + time stamps)	$\mathbf{9.6 \times 10^{-3}}$	$\mathbf{2.9 \times 10^{-5}}$	$\mathbf{5.4 \times 10^{-3}}$

Table 3: Enrichment of dataset-specific biological pathways for Velorama-inferred transcriptional regulators from single-cell time series data. Integrating time stamps alongside pseudotime with Velorama contributes to more significant enrichment of relevant pathways compared to Velorama with pseudotime alone.

4 Discussion

scRNA-seq datasets capture rich information about gene regulatory dynamics, parts of which are neglected by current techniques due to their inability to operate on partial (nonlinear) orderings of cells. This is particularly a problem for RNA velocity data, since many existing methods require a pseudotime estimate. An alternative approach is described in Scribe [9] where RNA velocity is considered, in essence, as two time-points of scRNA-seq data: one corresponding to spliced counts (regulators) and the other to unspliced counts (targets). However, this makes the heavily-questioned assumption [13] that splice rates across genes are equal, is not designed to identify interactions spanning multiple cell states, and is unable to take advantage of cell transition estimates offered by methods like scVelo and CellRank [14, 37]. Velorama takes a conceptually different approach by modeling the differentiation dynamics suggested by RNA velocity as a DAG. Velorama is also compatible with pseudotime if RNA velocity is unavailable. While our previous work introduced DAG-based representations of cellular dynamics [17], Velorama is the first to integrate per-cell transition probabilities from RNA velocity for GRN inference. Furthermore, Velorama is also the first method to leverage these DAG-based dynamics to evaluate multiple candidate regulators simultaneously along with their combinatorial effects.

Velorama's effectiveness on the GRN inference task opens the door to several directions for future work. Firstly, while Velorama has demonstrated notable performance using only scRNA-seq data, integrating this data with other data modalities, like chromatin accessibility and/or TF binding data, may contribute to further performance gains [29]. In addition, Velorama's ability to consider numerous regulators simultaneously allows for further investigation into the mechanisms underlying important co-regulatory relationships that underpin coordinated gene expression control [45, 46]. Lastly, we have demonstrated the effectiveness of using Velorama with either pseudotime or RNA velocity, but opportunities exist to integrate these two approaches. By ensembling cell-cell similarity predictions from these two approaches, we can potentially harness the unique features captured by both to more faithfully represent the underlying cellular dynamics used by Velorama for GRN inference.

Velorama unlocks crucial information from RNA velocity that otherwise would be lost with other techniques. As efforts to model gene expression dynamics from RNA velocity continue to grow and improve, Velorama's distinctive ability to flexibly leverage RNA velocity for causal GRN inference positions it to be an invaluable tool for uncovering novel mechanistic insights into differentiation and disease.

Acknowledgements AW, RS and BB were supported by the NIH grant R35GM141861. Some figures were made using biorender.com.

Declaration of Interests None

References

[1] M. d. Hoon, S. Imoto, and S. Miyano, "Inferring gene regulatory networks from time-ordered gene expression data using differential equations," in *International Conference on Discovery Science*, pp. 267–274, Springer, 2002.

[2] A. J. Hartemink, "Reverse engineering gene regulatory networks," *Nature biotechnology*, vol. 23, no. 5, pp. 554–555, 2005.

[3] J. Ding, N. Sharon, and Z. Bar-Joseph, "Temporal modelling using single-cell transcriptomics," *Nature Reviews Genetics*, vol. 23, no. 6, pp. 355–368, 2022.

[4] B. Hie, J. Peters, S. K. Nyquist, A. K. Shalek, B. Berger, and B. D. Bryson, "Computational methods for single-cell rna sequencing," *Annual Review of Biomedical Data Science*, vol. 3, pp. 339–364, 2020.

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

[6] 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.

[7] 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.

[8] 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.

[9] X. Qiu, A. Rahimzamani, L. Wang, B. Ren, Q. Mao, T. Durham, J. L. McFaline-Figueroa, L. Saunders, C. Trapnell, and S. Kannan, "Inferring causal gene regulatory networks from coupled single-cell expression dynamics using scribe," *Cell systems*, vol. 10, no. 3, pp. 265–274, 2020.

[10] 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.

[11] A. Deshpande, L.-F. Chu, R. Stewart, and A. Gitter, "Network inference with granger causality ensembles on single-cell transcriptomics," *Cell reports*, vol. 38, no. 6, p. 110333, 2022.

[12] G. La Manno, R. Soldatov, A. Zeisel, E. Braun, H. Hochgerner, V. Petukhov, K. Lidschreiber, M. E. Kastriti, P. Lönnerberg, A. Furlan, *et al.*, "Rna velocity of single cells," *Nature*, vol. 560, no. 7719, pp. 494–498, 2018.

[13] V. Bergen, R. A. Soldatov, P. V. Kharchenko, and F. J. Theis, "Rna velocity—current challenges and future perspectives," *Molecular systems biology*, vol. 17, no. 8, p. e10282, 2021.

[14] M. Lange, V. Bergen, M. Klein, M. Setty, B. Reuter, M. Bakhti, H. Lickert, M. Ansari, J. Schniering, H. B. Schiller, *et al.*, "Cellrank for directed single-cell fate mapping," *Nature methods*, vol. 19, no. 2, pp. 159–170, 2022.

[15] 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, p. 4774–4780, 2020.

[16] R. Liu, A. O. Pisco, E. Braun, S. Linnarsson, and J. Zou, "Dynamical systems model of rna velocity improves inference of single-cell trajectory, pseudo-time and gene regulation," *Journal of Molecular Biology*, vol. 434, no. 15, p. 167606, 2022.

[17] R. Singh, A. P. Wu, and B. Berger, "Granger causal inference on dags identifies genomic loci regulating transcription," in *International Conference on Learning Representations*, 2021.

[18] J. Kim, M. Choi, J.-R. Kim, H. Jin, V. N. Kim, and K.-H. Cho, "The co-regulation mechanism of transcription factors in the human gene regulatory network," *Nucleic Acids Research*, vol. 40, no. 18, p. 8849–8861, 2012.

[19] P. Dibaeinia and S. Sinha, "Sergio: a single-cell expression simulator guided by gene regulatory networks," *Cell systems*, vol. 11, no. 3, pp. 252–271, 2020.

[20] C. Hiemstra and J. D. Jones, "Testing for linear and nonlinear granger causality in the stock price-volume relation," *The Journal of Finance*, vol. 49, no. 5, pp. 1639–1664, 1994.

[21] A. K. Seth, A. B. Barrett, and L. Barnett, "Granger causality analysis in neuroscience and neuroimaging," *Journal of Neuroscience*, vol. 35, no. 8, pp. 3293–3297, 2015.

[22] A. Fujita, J. R. Sato, M. Garay, R. Yamaguchi, S. Miyano, M. Sogayar, and C. E. Ferreira, "Modeling gene expression regulatory networks with the sparse vector autoregressive model," *BMC Systems Biology*, vol. 1(39), 2007.

[23] S. Yao, S. Yoo, and D. Yu, "Prior knowledge driven granger causality analysis on gene regulatory network discovery," *BMC Bioinformatics*, vol. 16(273), 2015.

[24] F. Benhmad, "Modeling nonlinear Granger causality between the oil price and US dollar: A wavelet based approach," *Economic modelling*, vol. 29, no. 4, pp. 1505–1514, 2012.

[25] A. Tank, I. Covert, N. Foti, A. Shojaie, and E. B. Fox, "Neural granger causality," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[26] R. Marcinkevičs and J. E. Vogt, "Interpretable models for granger causality using self-explaining neural networks," *arXiv preprint arXiv:2101.07600*, 2021.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[28] N. Parikh, S. Boyd, *et al.*, "Proximal algorithms," *Foundations and trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[29] R. Singh, B. Hie, A. Narayan, and B. Berger, "Schema: metric learning enables interpretable synthesis of heterogeneous single-cell modalities," *Genome Biology*, vol. 22, no. 131, 2021.

[30] K. Hara, D. Saito, and H. Shouno, "Analysis of function of rectified linear unit used in deep learning," in *2015 international joint conference on neural networks (IJCNN)*, pp. 1–8, IEEE, 2015.

[31] B. Hie, H. Cho, B. DeMeo, B. Bryson, and B. Berger, "Geometric sketching compactly summarizes the single-cell transcriptomic landscape," *Cell systems*, vol. 8, no. 6, pp. 483–493, 2019.

[32] B. DeMeo and B. Berger, "Hopper: a mathematically optimal algorithm for sketching biological data," *Bioinformatics*, vol. 36, pp. i236–i241, 07 2020.

[33] A. Bastidas-Ponce, S. Tritschler, L. Dony, K. Scheibner, M. Tarquis-Medina, C. Salinno, S. Schirge, I. Burtscher, A. Böttcher, F. Theis, and et al., "Comprehensive single-cell mrna profiling reveals a detailed roadmap for pancreatic endocrinogenesis," *Development*, vol. 146, no. 12, 2019.

[34] H. Hochgerner, A. Zeisel, P. Lönnerberg, and S. Linnarsson, "Conserved properties of dentate gyrus neurogenesis across postnatal development revealed by single-cell rna sequencing," *Nature Neuroscience*, vol. 21, no. 2, p. 290–299, 2018.

[35] M. Setty, V. Kiseliovas, J. Levine, A. Gayoso, L. Mazutis, and D. Pe'er, "Characterization of cell fate probabilities in single-cell data with Palantir," *Nat Biotechnol*, vol. 37, pp. 451–460, 04 2019.

[36] F. A. Wolf, P. Angerer, and F. J. Theis, "Scanpy: large-scale single-cell gene expression data analysis," *Genome biology*, vol. 19, no. 1, pp. 1–5, 2018.

[37] V. Bergen, M. Lange, S. Peidli, F. A. Wolf, and F. J. Theis, "Generalizing rna velocity to transient cell states through dynamical modeling," *Nature biotechnology*, vol. 38, no. 12, pp. 1408–1414, 2020.

[38] Z. Ren, P. Yu, D. Li, Z. Li, Y. Liao, Y. Wang, B. Zhou, and L. Wang, "Single-cell reconstruction of progression trajectory reveals intervention principles in pathological cardiac hypertrophy," *Circulation*, vol. 141, no. 21, p. 1704–1719, 2020.

[39] S. R. Srivatsan, J. L. McFaline-Figueroa, V. Ramani, L. Saunders, J. Cao, J. Packer, H. A. Pliner, D. L. Jackson, R. M. Daza, L. Christiansen, and et al., "Massively multiplex chemical transcriptomics at single-cell resolution," *Science*, vol. 367, no. 6473, p. 45–51, 2020.

[40] 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.

[41] E. Y. Chen, C. M. Tan, Y. Kou, Q. Duan, Z. Wang, G. V. Meirelles, N. R. Clark, and A. Ma'ayan, "Enrichr: Interactive and collaborative html5 gene list enrichment analysis tool," *BMC Bioinformatics*, vol. 14, no. 1, 2013.

[42] D. Lara-Astiaso, A. Weiner, E. Lorenzo-Vivas, I. Zaretsky, D. A. Jaitin, E. David, H. Keren-Shaul, A. Mildner, D. Winter, S. Jung, N. Friedman, and I. Amit, "Chromatin state dynamics during blood formation," *Science*, vol. 345, no. 6199, pp. 943–949, 2014.

[43] R. Ostuni, V. Piccolo, I. Barozzi, S. Polletti, A. Termanini, S. Bonifacio, A. Curina, E. Prosperini, S. Ghisletti, and G. Natoli, "Latent enhancers activated by stimulation in differentiated cells," *Cell*, vol. 1-2, pp. 157–171, 2013.

[44] J. Ding, B. J. Aronow, N. Kaminski, J. Kitzmiller, J. A. Whitsett, and Z. Bar-Joseph, "Reconstructing differentiation networks and their regulation from time series single-cell expression data," *Genome Research*, vol. 28, no. 3, p. 383–395, 2018.

[45] B. Hie, H. Cho, B. Bryson, and B. Berger, "Coexpression enables multi-study cellular trajectories of development and disease," *bioRxiv*, p. 719088, 2020.

[46] J. Zhang, J. Liu, D. Lee, S. Lou, Z. Chen, G. Gürsoy, and M. Gerstein, "Diner: a differential graphical model for analysis of co-regulation network rewiring," *BMC bioinformatics*, vol. 21, no. 1, pp. 1–15, 2020.

[47] H. Hu, Y.-R. Miao, L.-H. Jia, Q.-Y. Yu, Q. Zhang, and A.-Y. Guo, "Animaltfdb 3.0: A comprehensive resource for annotation and prediction of animal transcription factors," *Nucleic Acids Research*, vol. 47, no. D1, 2018.

A Appendix

A.1 Velorama Architecture

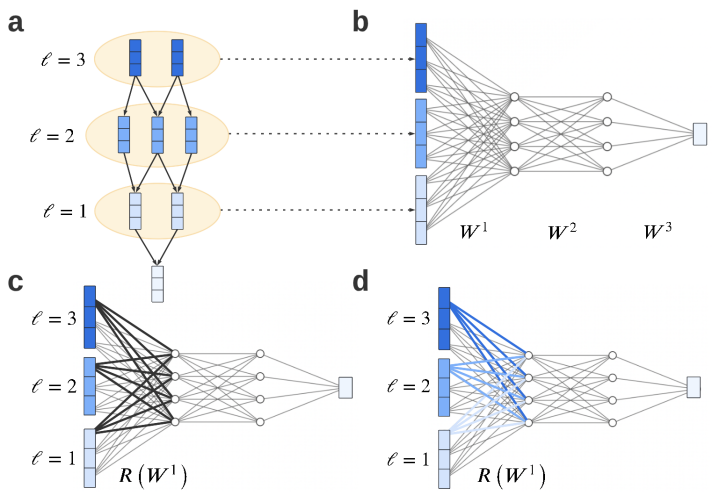

Figure A.1: Velorama architecture: $N = 8$, $G = 3$, $L = 3$, $d = 4$ and $K = 3$. (a) Velorama predicts a node's features based on ancestors in the DAG. (b) Predictions are made by passing aggregated information from each lag through a feed-forward neural network. (c) If the weights associated with variable i in f_j (highlighted in black for $i = 1$) are all equal to 0, then variable i does *not* Granger-cause variable j. (d) The lag-selection and hierarchical penalties further penalize groups of weights associated with particular lags.

A.2 Regularization Schemes

Here, $\mathbf{W}^{(j)}$ denotes the $\mathbf{W}^1$ corresponding to model f_j and $\mathbf{W} = [\mathbf{W}^{(1)} \| \dots \| \mathbf{W}^{(G)}] \in \mathbb{R}^{G \times L \times G \times d}$ represents the concatenation of these matrices over all G models.

The *group* regularization penalizes all weights within $\mathbf{W}$ symmetrically (**Figure** A.1c):

$$R(\mathbf{W}) = \sum_{i=1}^{G} \sqrt{\sum_{j=1}^{G} \left\| W_i^{(j)} \right\|_F^2}$$

The *lag-selection* regularization aids in automatic lag detection by merging weights from the same lag together (**Figure** A.1d):

$$R(\mathbf{W}) = \sum_{i=1}^{G} \left(\sqrt{\sum_{j=1}^{G} \left\| W_i^{(j)} \right\|_F^2} + \sum_{l=1}^{L} \sqrt{\sum_{j=1}^{G} \left\| \mathbf{W}_i^{(j),l} \right\|_2^2} \right)$$

A.3 Datasets and pre-processing

A.3.1 SERGIO

We applied Velorama to four benchmark synthetic datasets obtained from the SERGIO GitHub repository (**Table** A.1). Prior to running Velorama, we normalized the expression values of each gene to have a mean of 0 and a standard deviation of 1. To evaluate Velorama, we compared the inferred causal graph to the ground-truth set of regulator–target pairs used to simulate the expression matrix. All 100 genes were considered as candidate regulators and targets.

	Differentiation	Source
Dataset **A**	Linear	`https://github.com/PayamDiba/SERGIO/tree/master/data_sets/De-noised_100G_3T_300cPerT_dynamics_9_DS4`
Dataset **B**	Bifurcation	`https://github.com/PayamDiba/SERGIO/tree/master/data_sets/De-noised_100G_4T_300cPerT_dynamics_10_DS5`
Dataset **C**	Trifurcation	`https://github.com/PayamDiba/SERGIO/tree/master/data_sets/De-noised_100G_6T_300cPerT_dynamics_7_DS6`
Dataset **D**	Tree-shaped	`https://github.com/PayamDiba/SERGIO/tree/master/data_sets/De-noised_100G_7T_300cPerT_dynamics_11_DS7`

Table A.1: SERGIO dataset download links

A.3.2 RNA velocity and time series datasets

We analyzed five scRNA-seq datasets profiling mouse endocrinogenesis [33], mouse dentate gyrus neurogenesis [34], human hematopoiesis [35], cardiac hypertrophy in cardiomyocytes [38], and the response of A549 cells to pracinostat [39]. We performed the same pre-processing procedure for each of these datasets. We first normalized the number of counts per cell such that each cell has a total count equal to the pre-normalization median number of counts per cell. We then applied a $\log(1 + X)$ transformation on the normalized counts X and identified the top 2000 most highly variable genes. The transformed expression values for each gene were then normalized to have a mean of 0 and a standard deviation of 1. We removed all genes that were represented in fewer than 1% of cells in each dataset and retained the union of the remaining highly variable genes and genes that were annotated as transcription factors according to AnimalTFDB [47]. When applying Velorama, transcription factors were selected as the set of candidate regulators, and candidate target genes were defined to be the set of highly variable genes excluding transcription factors.

A.4 Runtimes of GRN inference algorithms on SERGIO datasets

A.5 Velorama integration of time stamps from scRNA-seq time series datasets

Temporal information from scRNA-seq time series data can be flexibly integrated with Velorama to better inform temporal relationships between single cell measurements. scRNA-seq time series experiments typically profile populations of cells at multiple distinct time points, such that each profiled cell is associated with a coarse time stamps. We use the single-cell multimodal integration algorithm Schema [29] to endow the cells' gene expression representations with information from these time stamps. Specifically, we use Schema to synthesize the PCA representations of cells' expression profiles with the datasets' time stamps, treating the former as the primary modality and the latter as the secondary modality in the Schema workflow. We set the minimum correlation between the primary and secondary modalities to be 0.9. Schema generates transformed cell representations that unify the gene expression and time stamp modalities, and we use these new cell representations for constructing the Velorama DAG.

Method	Dataset **A**	Dataset **B**	Dataset **C**	Dataset **D**
SINGE	49.9	87.5	183.9	248.7
SINCERITIES	0.2	0.2	0.2	0.2
SCODE	9.1	10.4	13.2	14.5
PIDC	0.3	0.3	0.4	0.4
Scribe	3.9	5.3	9.2	12.9
GRNBoost2	3.3	5.4	11.5	10.3
GENIE3	6.9	9.2	14.1	17.3
RNA-ODE	12.9	17.3	25.5	32.5
Velorama	2.5	2.6	3.3	2.9

Table A.2: GRN inference runtimes (in minutes)

Method	Dataset **A**		Dataset **B**		Dataset **C**		Dataset **D**	
	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC
Velorama (pseudotime, $d = 8$)	0.172	0.823	0.153	0.846	0.416	0.886	0.325	0.845
Velorama (pseudotime, $d = 16$)	0.191	0.848	0.154	0.856	0.327	0.882	0.296	0.872
Velorama (pseudotime, $d = 32$)	0.184	0.871	0.175	0.874	0.390	0.892	0.325	0.846
Velorama (velocity, $d = 8$)	0.404	0.905	0.498	0.855	0.590	0.919	0.588	0.893
Velorama (velocity, $d = 16$)	0.399	0.900	0.516	0.856	0.596	0.923	0.541	0.926
Velorama (velocity, $d = 32$)	0.390	0.904	0.560	0.884	0.539	0.937	0.653	0.872

Table A.3: Velorama performance on SERGIO datasets across different choices of number of hidden dimensions d

A.6 Velorama robustness to hyperparameter choices

Method	Dataset **A**		Dataset **B**		Dataset **C**		Dataset **D**	
	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC
Velorama (velocity, stochastic)	0.399	0.900	0.516	0.856	0.596	0.923	0.541	0.926
Velorama (velocity, dynamical)	0.388	0.900	0.387	0.878	0.556	0.914	0.469	0.858

Table A.6: Velorama performance on SERGIO datasets across different choices of RNA velocity dynamical models

A.7 Velorama-inferred GRNs for SERGIO datasets

We plotted the ground-truth GRN as well as the Velorama-inferred GRNs using pseudotime and RNA velocity for datasets A, B and C (**Figure** A.2). Using RNA velocity improves predictions of the GRN.

Method	Dataset **A**		Dataset **B**		Dataset **C**		Dataset **D**	
	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC
Velorama (pseudotime, $k = 15$)	0.191	0.848	0.154	0.856	0.327	0.882	0.296	0.872
Velorama (pseudotime, $k = 30$)	0.191	0.848	0.168	0.862	0.384	0.885	0.323	0.845
Velorama (velocity, $k = 15$)	0.399	0.900	0.516	0.856	0.596	0.923	0.541	0.926
Velorama (velocity, $k = 30$)	0.399	0.900	0.556	0.880	0.489	0.942	0.552	0.925

Table A.4: Velorama performance on SERGIO datasets across different choices of number of nearest neighbors k

Method	Dataset **A**		Dataset **B**		Dataset **C**		Dataset **D**	
	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC
Velorama (pseudotime, $M = 30$)	0.127	0.875	0.137	0.852	0.161	0.893	0.394	0.879
Velorama (pseudotime, $M = 50$)	0.191	0.848	0.168	0.862	0.384	0.885	0.323	0.845
Velorama (velocity, $M = 30$)	0.381	0.888	0.521	0.866	0.484	0.932	0.501	0.940
Velorama (velocity, $M = 50$)	0.399	0.900	0.556	0.880	0.489	0.942	0.552	0.925

Table A.5: Velorama performance on SERGIO datasets across different choices of number of PCs M

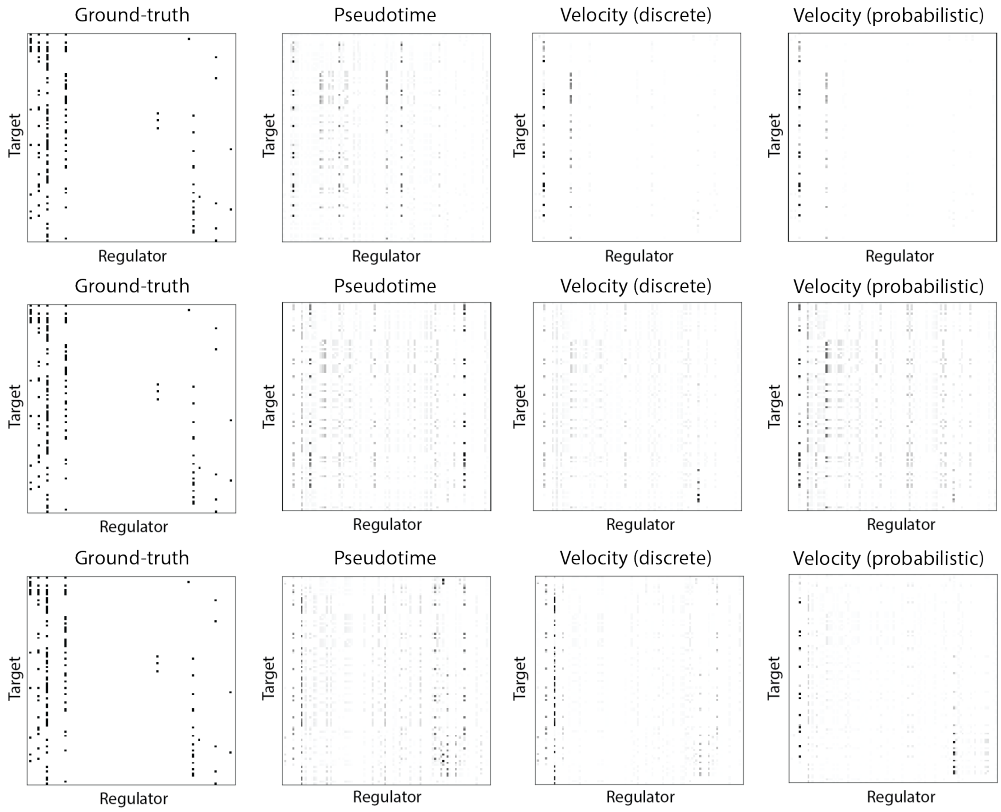

Figure A.2: Ground-truth GRN for Datasets **A** (top), **B** (middle) and **C** (bottom) compared to the Velorama-inferred GRNs when using pseudotime or RNA velocity data.

A.8 Velorama time stamp integration and GRN inference accuracy on SERGIO datasets

SERGIO's default settings do not produce time stamps for each cell, but we can manually assign time stamps that cohere with SERGIO's procedure for simulating differentiation trajectories. For a given differentiation trajectory, a cell type transition is initiated by changing the production rates of the underlying GRN's master regulators, after which SERGIO's time-course simulations are applied to generate the downstream expression changes that follow from these modified rates. As a result, the cell type transition points demarcate the boundaries of the distinct time windows that are used in the simulation, and each cell type's expression profiles directly corresponds to one of these time windows. For a linear trajectory like Dataset A, each cell type would be assigned a distinct time stamp ($t = 0, 1, 2$) (**Figure** 2). For branching trajectories, the cell types that are differentiated from the same parent cell type exist at the same time point, and we mark each cell from a given cell type with the same time stamp. For example, for Dataset C, the time stamps assigned to each cell type would be as follows: red ($t = 0$), green ($t = 1$), blue ($t = 2$), orange ($t = 3$), purple ($t = 3$), yellow ($t = 3$) (**Figure** 2).

Method	Dataset **A**		Dataset **B**		Dataset **C**		Dataset **D**	
	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC
Velorama (pseudotime)	0.191	0.848	0.154	0.856	0.327	0.882	0.296	0.872
Velorama (pseudotime + time stamps)	0.112	0.894	0.482	0.912	0.592	0.951	0.672	0.939
Velorama (velocity)	0.399	0.900	0.516	0.856	0.596	0.923	0.541	0.926
Velorama (velocity + time stamps)	0.298	0.901	0.717	0.952	0.691	0.963	0.738	0.958

Table A.7: Velorama performance on SERGIO datasets with and without integrating time stamps

A.9 Velorama enables TF-target interaction speed predictions

Velorama's ability to perform lag selection enables it to predict the interaction speed for each regulator–target gene pair. For a target gene j, we quantify the Velorama interaction speed between a regulator i and the target by first scoring each of its associated lags l as the ℓ_2-norm of the weights $\mathbf{W}_i^{1,l} \in \mathbb{R}^d$. We obtain these scores for each λ that is associated with more than 1% non-zero lag scores across all inferred interactions and lags. We then binarize these scores, setting these scores to 1 if they are non-zero and 0 otherwise. We average these binarized scores across the retained λ values to yield a per-lag score $C_{ij}^{(l)}$ for each regulator-target gene pair (i, j). These values are ℓ_1-normalized over the lags for each regulator-target gene pair (i, j) to help account for differences in overall regulator strengths. Finally, we represent the interaction speed s_{ij} for a regulator i and target gene j by using these normalized scores to weight the contribution of each lag point l in the weighted sum below. We note that each lag point's speed is defined to be $(L - l + 1)$, so that the shortest lag $l = 1$ is associated with the fastest speed and the longest lag $l = L$ is associated with the slowest.

$$s_{ij} = \sum_{l=1}^{L} \tilde{C}_{ij}^{(l)} (L - l + 1), \text{ where } \tilde{C}_{ij} = \frac{C_{ij}}{\|C_{ij}\|_1} \text{ and } C_{ij} \in \mathbb{R}^L$$