

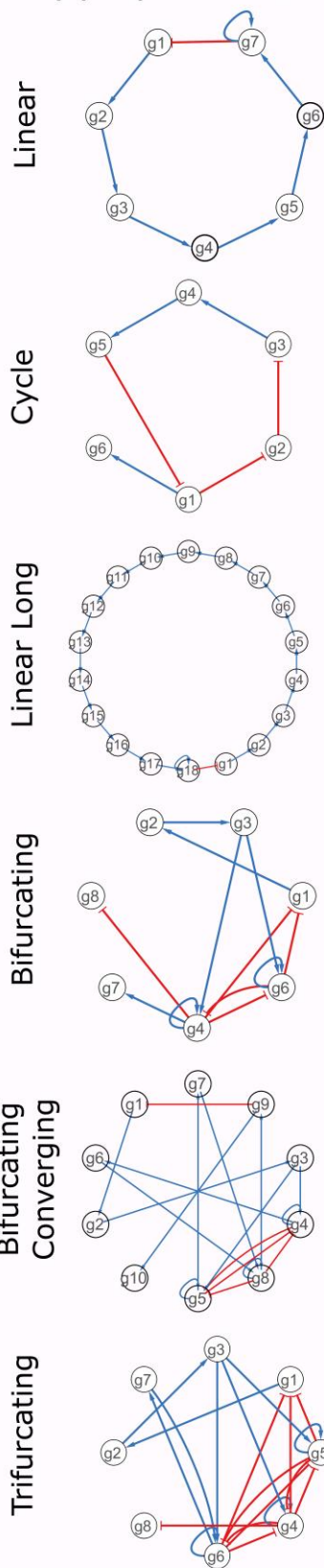
In the format provided by the authors and unedited.

Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data

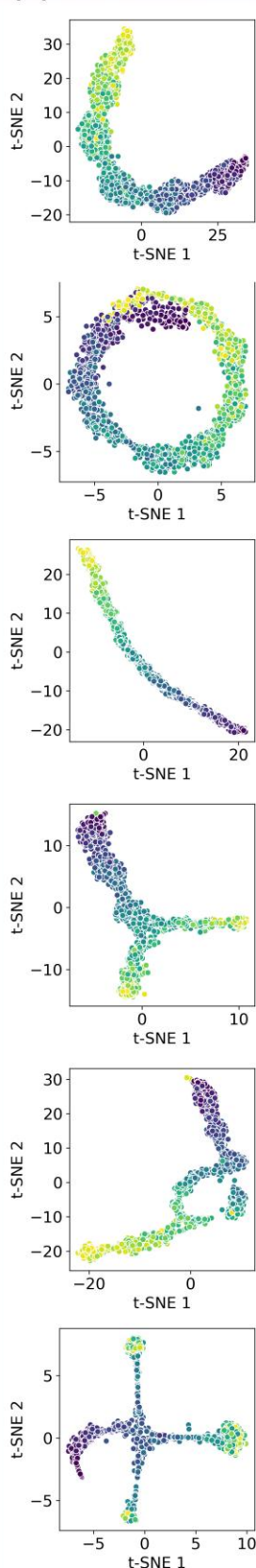
Aditya Pratapa ¹, Amogh P. Jaliha ², Jeffrey N. Law ², Aditya Bharadwaj¹ and T. M. Murali ^{1*}

¹Department of Computer Science, Virginia Tech, Blacksburg, VA, USA. ²Genetics, Bioinformatics, and Computational Biology Ph.D. Program, Virginia Tech, Blacksburg, VA, USA. *e-mail: murali@cs.vt.edu

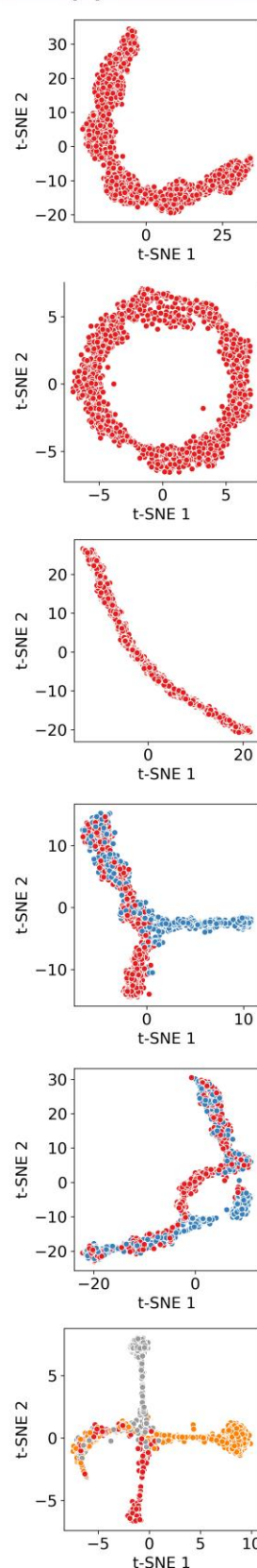
(a) Toy Networks



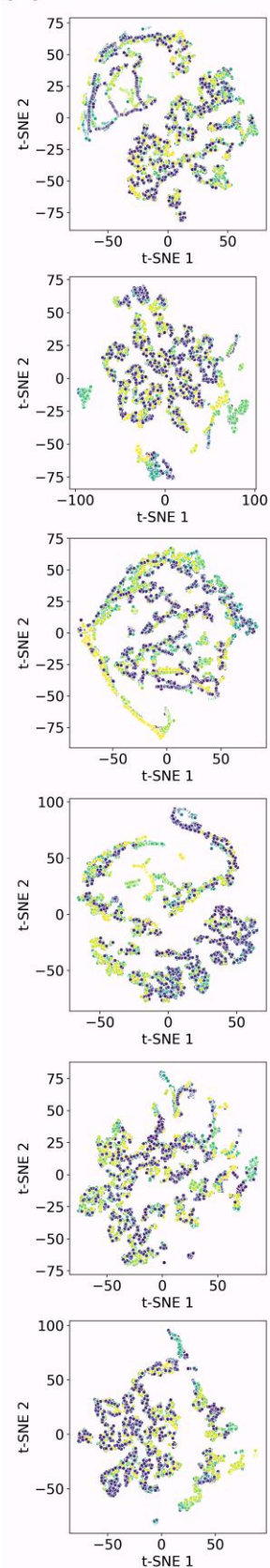
(b) Simulation Time



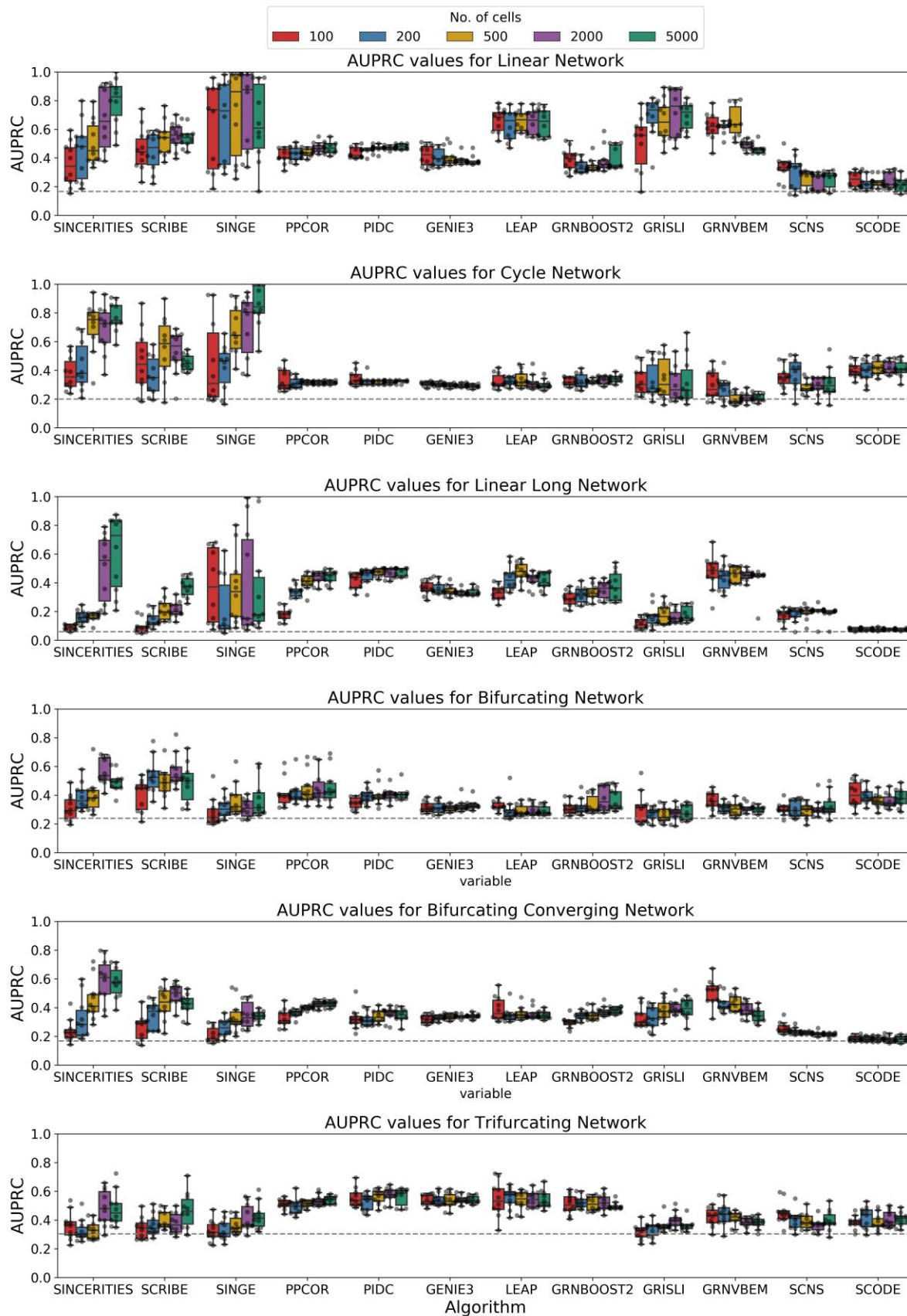
(c) Clusters



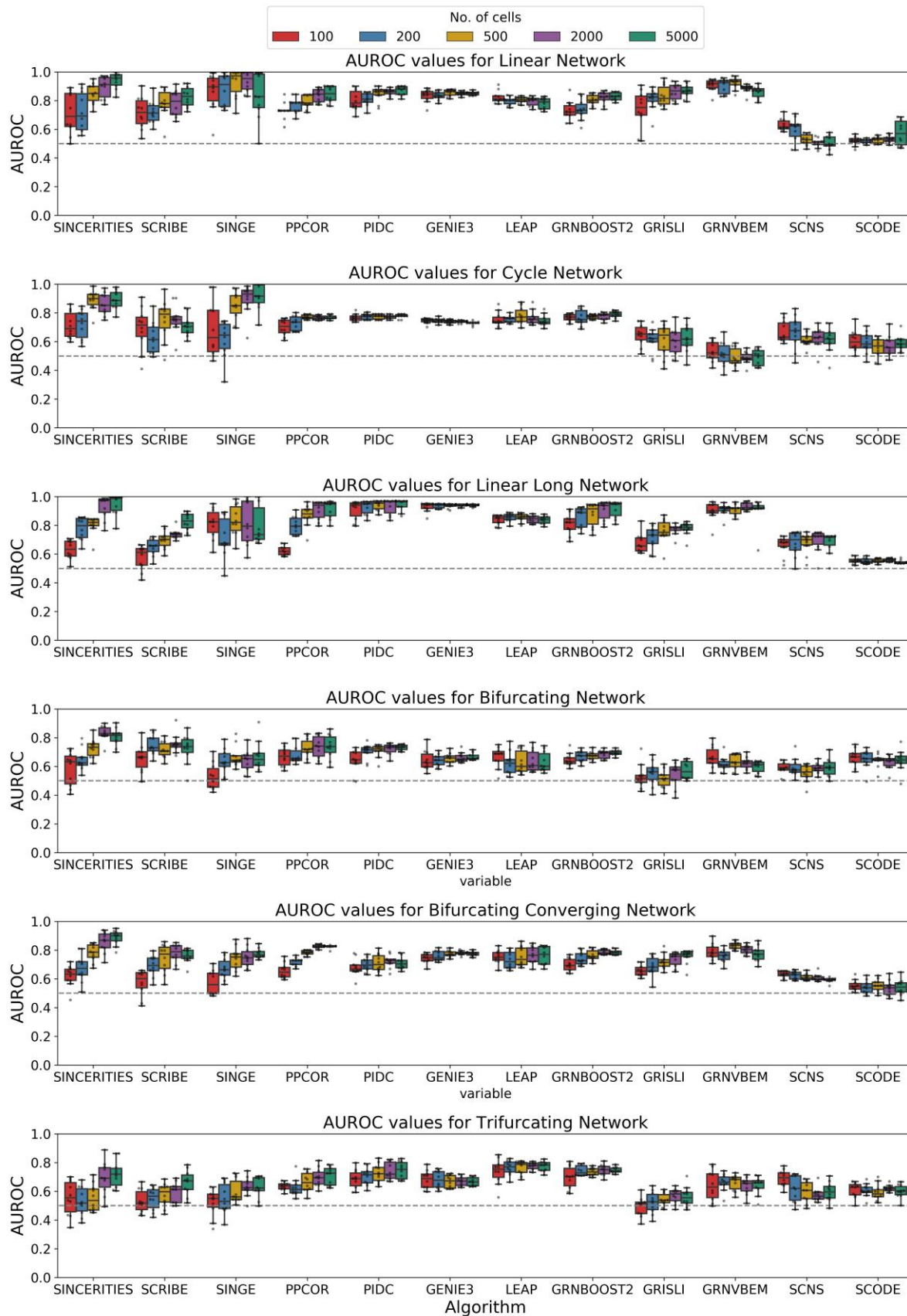
(d) GeneNetWeaver



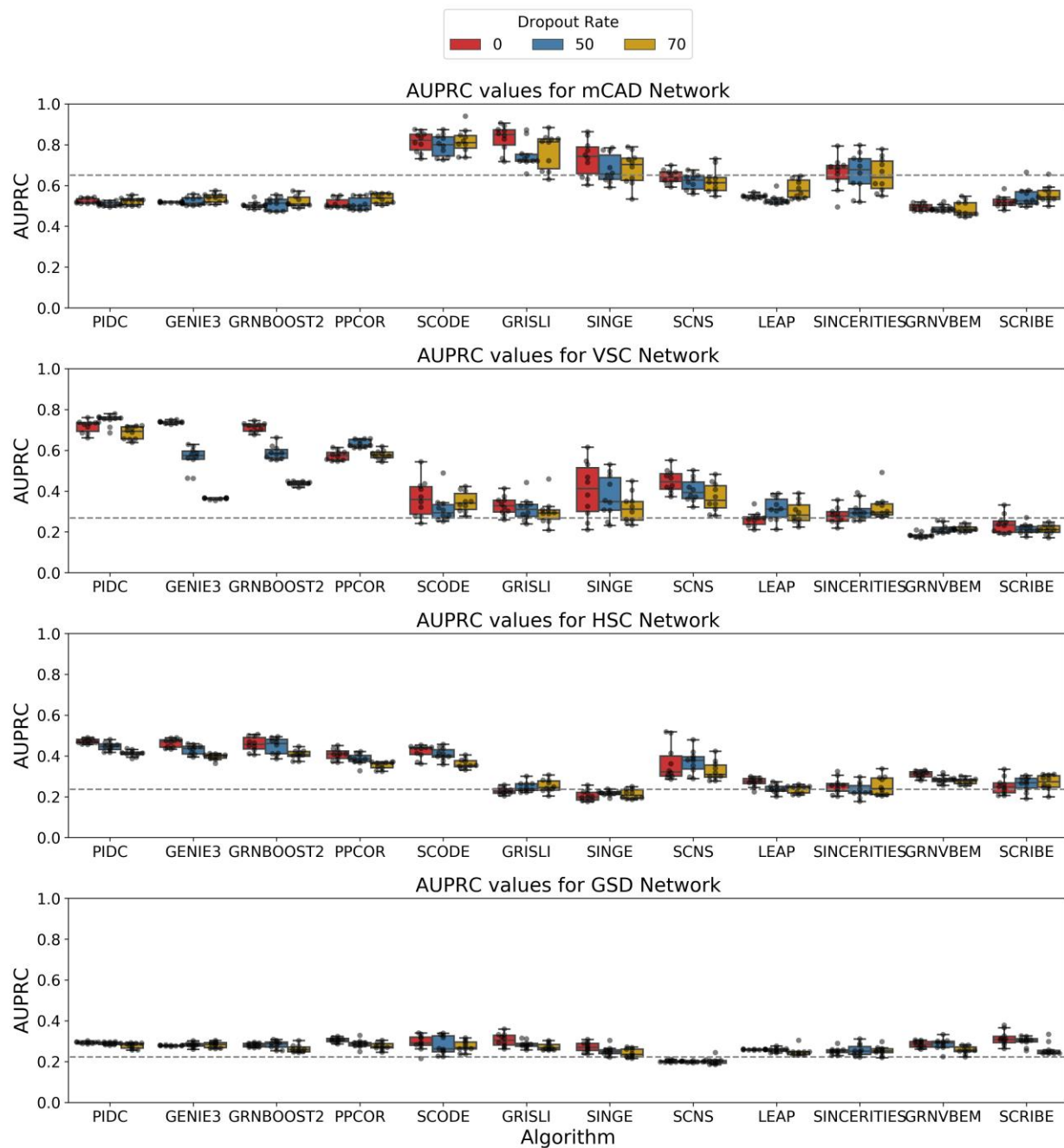
Supplementary Figure 1	
Comparison of datasets simulated from synthetic networks by using BoolODE and GeneNetWeaver.	
	Each row corresponds to the synthetic network indicated by the label on the left. (a) The network itself, with red edges representing inhibition and blue edges representing activation. (b) A 2D t-SNE visualization of one BoolODE-generated dataset for 2,000 cells. The color of each point indicates the simulation time: blue for earlier, green for intermediate, and yellow for later times. (c) Each colour corresponds to a different subset of cells obtained by using k-means clustering of the BoolODE-generated dataset, with k set to the number of expected steady states. (d) A 2-D t-SNE visualization of one GeneNetWeaver output.



Supplementary Figure 2
Box plots of AUPRC values for synthetic networks.
Each row corresponds to one of the six synthetic networks. Each column corresponds to an algorithm. Red, blue, yellow, purple and green box plots correspond to AUPRC values for 10 datasets with 100, 200, 500, 2,000, and 5,000 cells, respectively. The gray dotted line indicates the AUPRC value for a random predictor, which is equal to the network's density. In every boxplot, the box shows the 1 st and 3 rd quartile, and whiskers denote 1.5 times the interquartile range.



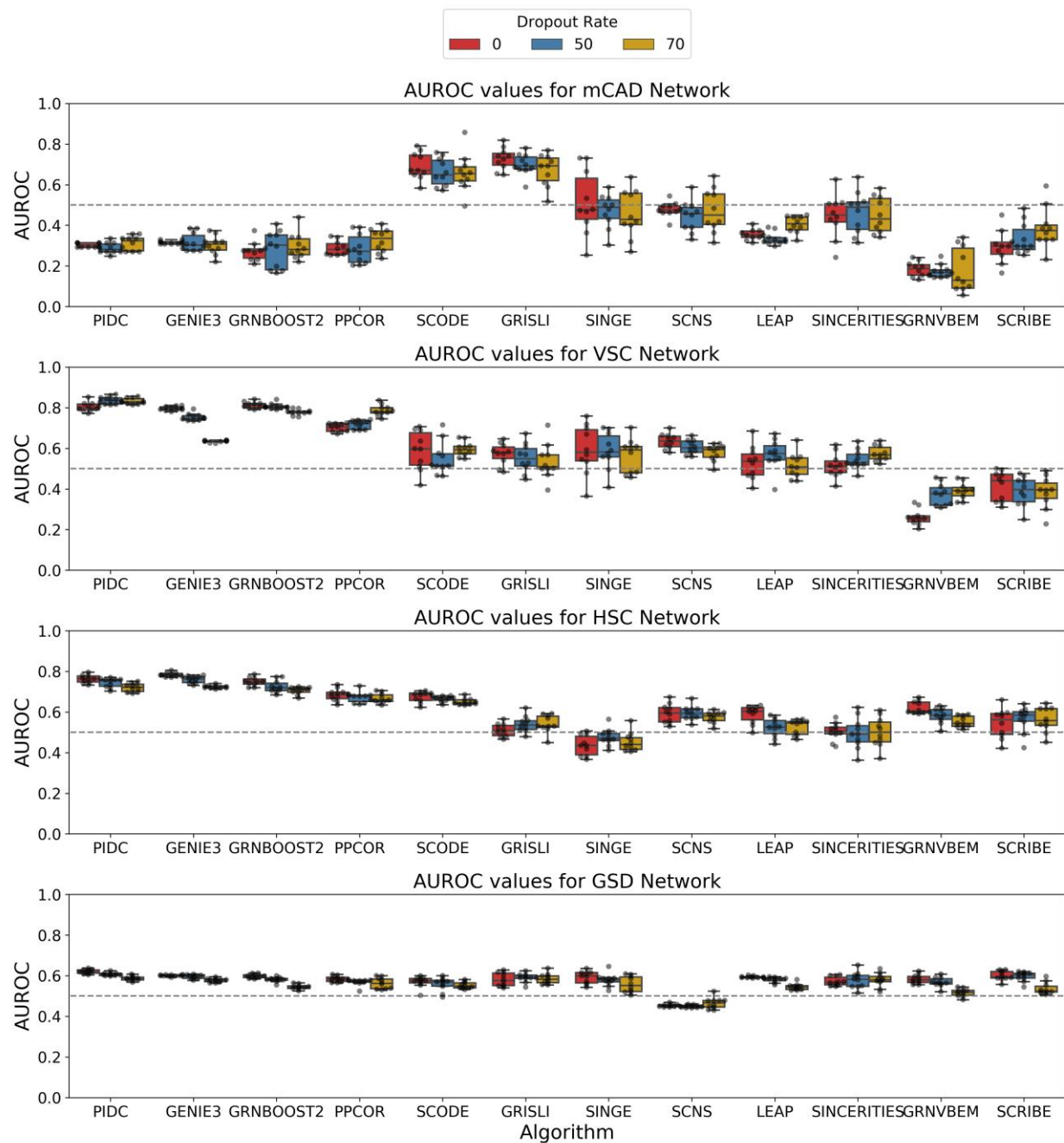
Supplementary Figure 3
Box plots of AUROC values for synthetic networks.
Each row corresponds to one of the six synthetic networks. Each column corresponds to an algorithm. Red, blue, yellow, purple and green box plots correspond to AUROC values for 10 datasets with 100, 200, 500, 2,000, and 5,000 cells, respectively. The gray dotted line indicates the AUROC value for a random predictor (0.5). In every boxplot, the box shows the 1 st and 3 rd quartile, and whiskers denote 1.5 times the interquartile range.



Supplementary Figure 4

Box plots of AUPRC values for curated models.

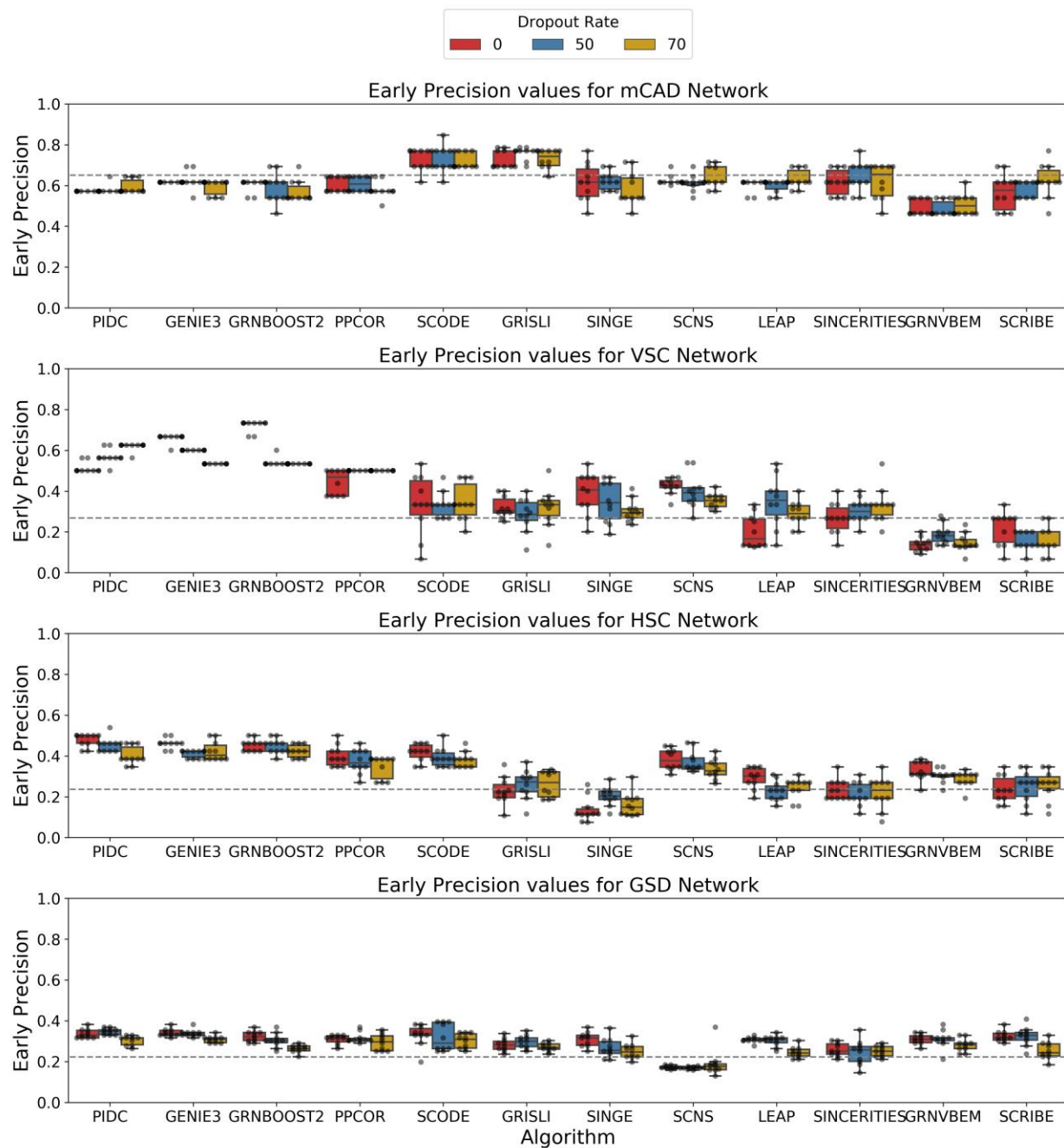
Each row corresponds to one of the four curated models. Each column corresponds to an algorithm. Red, blue and yellow box plots correspond to AUPRC values for 10 datasets with no dropouts, a dropout rate of $q = 50$, and a dropout rate of $q = 70$, respectively. The gray dotted line indicates the AUPRC value for a random predictor, i.e., the network density. In every boxplot, the box shows the 1st and 3rd quartile, and whiskers denote 1.5 times the interquartile range.



Supplementary Figure 5

Box plots of AUROC values for curated models.

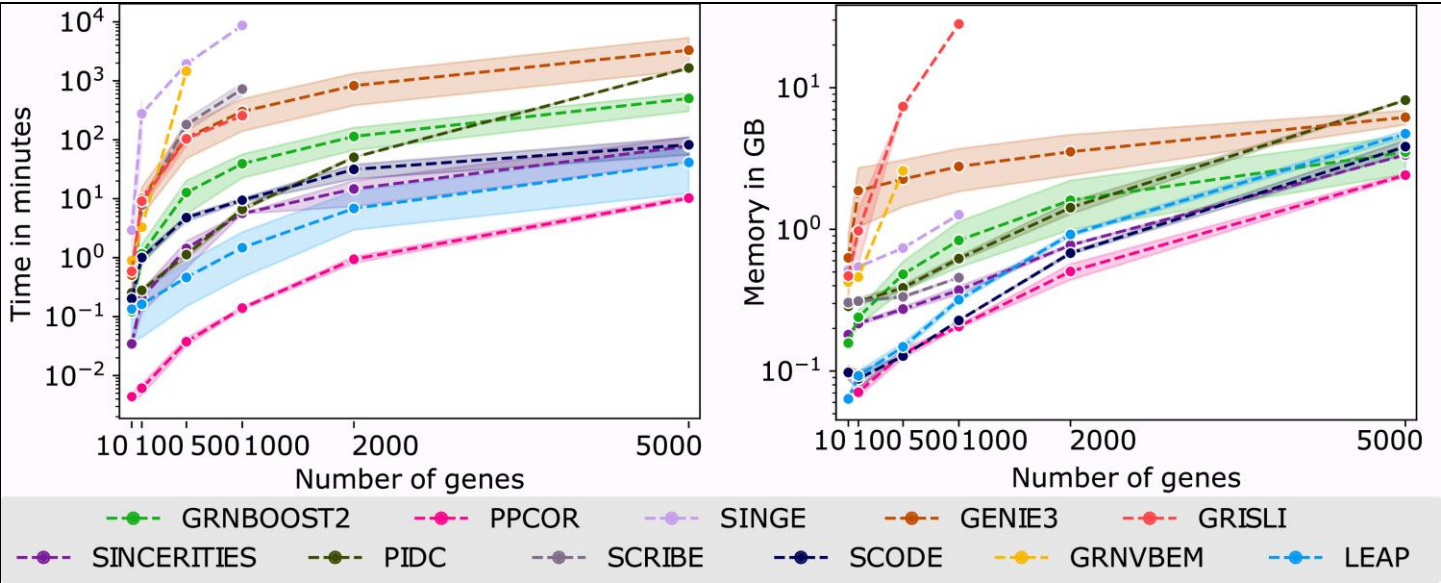
Each row corresponds to one of the four curated models. Each column corresponds to an algorithm. Red, blue and yellow box plots correspond to AUROC values for 10 datasets with no dropouts, a dropout rate of $q = 50$, and a dropout rate of $q = 70$, respectively. The gray dotted line indicates the AUROC value for a random predictor (0.5). In all boxplots, the box shows the 1st and 3rd quartile, and whiskers denote 1.5 times the interquartile range.



Supplementary Figure 6

Box plots of early precision values for curated models.

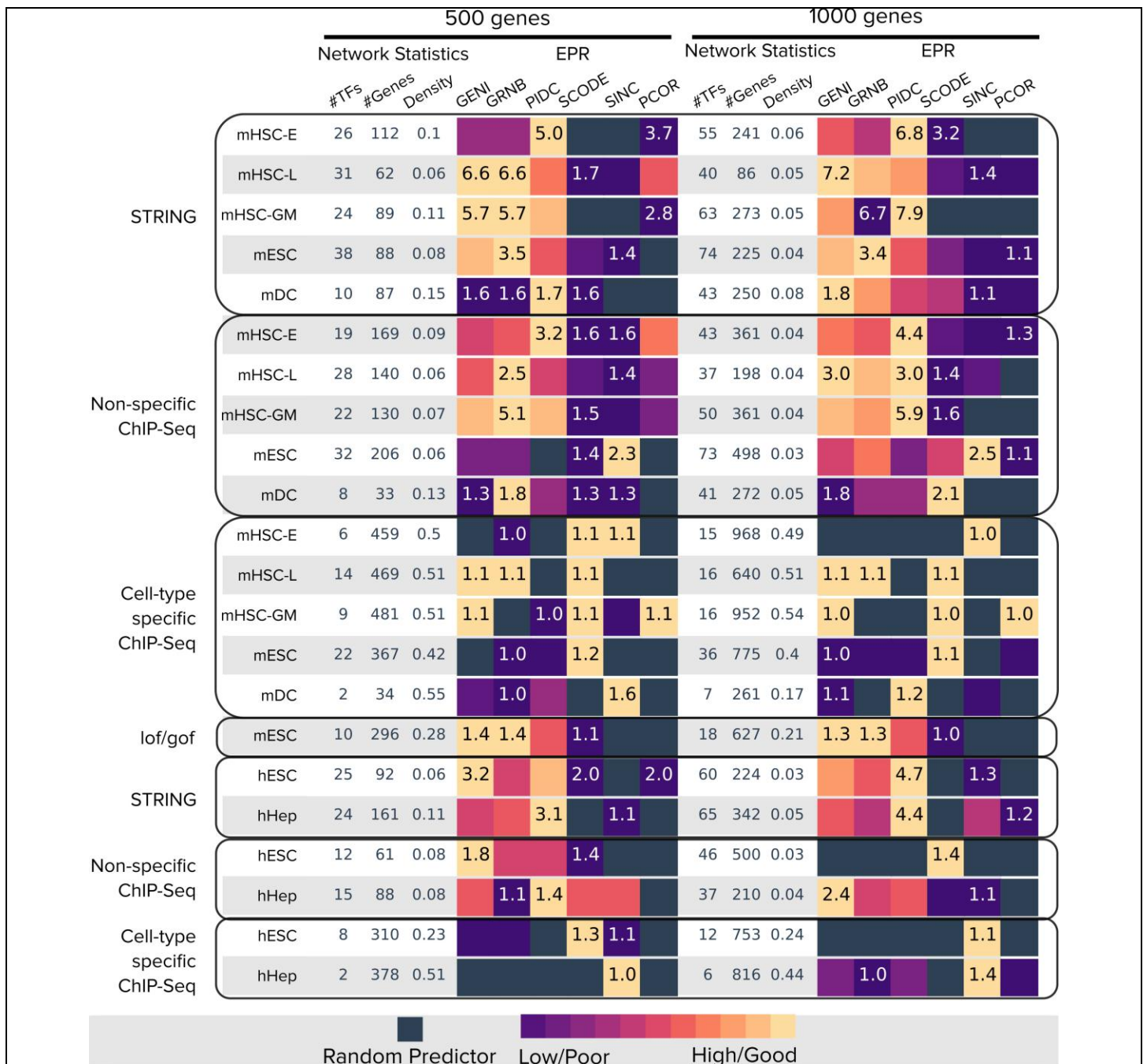
Each row corresponds to one of the four curated models. Each column corresponds to an algorithm. Red, blue and yellow box plots correspond to early precision values for 10 datasets with no dropouts, a dropout rate of $q = 50$, and a dropout rate of $q = 70$, respectively. The gray dotted line indicates the early precision value for a random predictor (network density). In each boxplot, the box shows the 1st and 3rd quartile, and whiskers denote 1.5 times the interquartile range.



Supplementary Figure 7

Scalability of GRN algorithms on experimental single-cell RNA-Seq datasets.

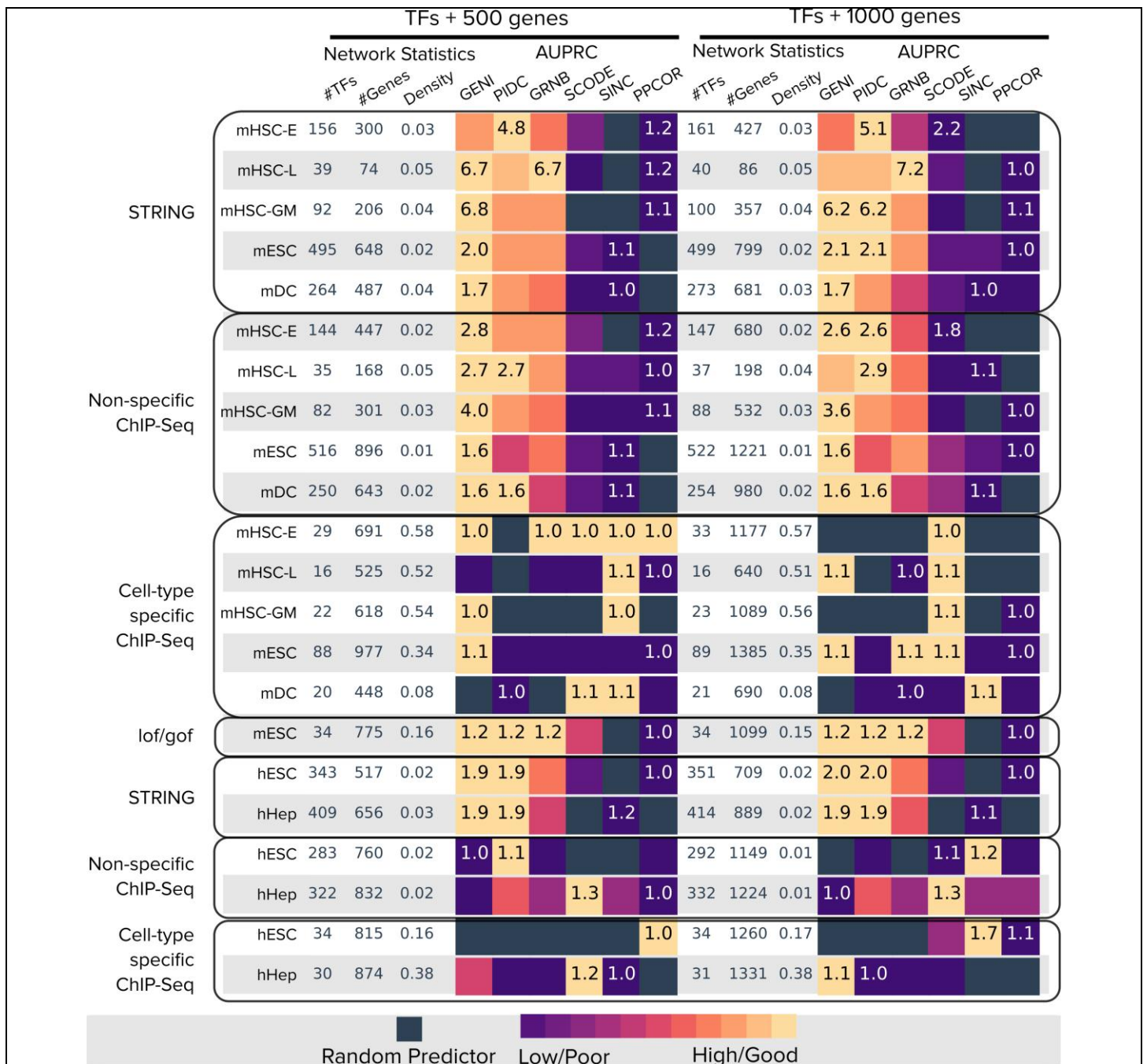
Variation in running time and memory usage of GRN inference algorithms with respect to number of genes for three experimental single-cell RNA-Seq datasets. Each point represents the mean running time or memory across all three datasets and the shaded regions correspond to one standard deviation around the mean. Missing values indicate that the method either did not complete after one day or gave a runtime error. We did not consider SCNS since it took over a day on the 19-gene GSD Boolean model. We obtained these results on a computer with a 32-core 2.0GHz processor and 32GB of memory running Ubuntu 18.04.



Supplementary Figure 8

Summary of EPR values for experimental single-cell RNA-Seq datasets with 500 and 1000 genes.

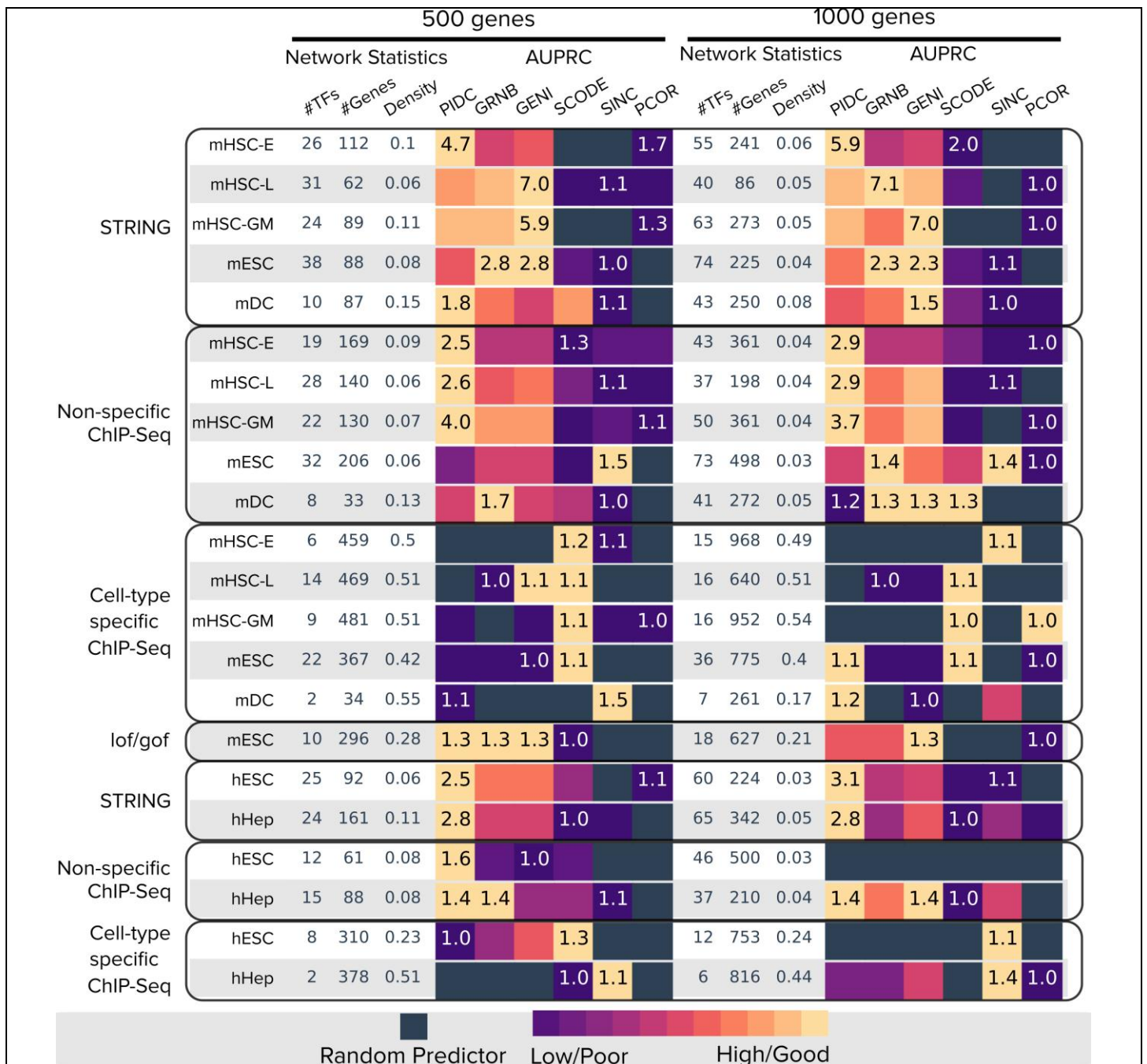
Summary of EPR results for experimental single-cell RNA-seq datasets. The left half of the figure (500 genes) shows results for datasets composed of the 500 most-varying genes. Each row corresponds to one scRNA-seq dataset. The first three columns report network statistics. The next six columns report EPR values. The right half (1000 genes) shows results for the 1000 most-varying genes. In both sections, algorithms are sorted by median EPR across the datasets (rows) for the 500 gene set. For each dataset, the color in each cell is proportional to the corresponding value scaled between 0 and 1 (ignoring values that are less than that of a random predictor, which are shown as black squares). We display the highest and lowest values for each dataset inside the corresponding cells. Abbreviations: GENI: GENIE3, GRNB: GRNBoost2, PCOR: PPCOR, SINC: SINCERITIES.



Supplementary Figure 9

Summary of AUPRC ratio values for experimental single-cell RNA-Seq datasets with TFs + 500 and TFs + 1000 genes.

Summary of AUPRC ratio results for experimental single-cell RNA-seq datasets. The left half of the figure (TFs+500 genes) shows results for datasets composed of all significantly-varying TFs and the 500 most-varying genes. Each row corresponds to one scRNA-seq dataset. The first three columns report network statistics. The next six columns report AUPRC ratios. The right half (TFs+1000 genes) shows results for all significantly-varying TFs and the 1000 most-varying genes. In both sections, algorithms are sorted by median AUPRC ratio across the datasets (rows) for the TFs+500 gene set. For each dataset, the color in each cell is proportional to the corresponding value scaled between 0 and 1 (ignoring values that are less than that of a random predictor, which are shown as black squares). We display the highest and lowest values for each dataset inside the corresponding cells. Abbreviations: GENI: GENIE3, GRNB: GRNBoost2, PCOR: PPCOR, SINC: SINCERITIES.

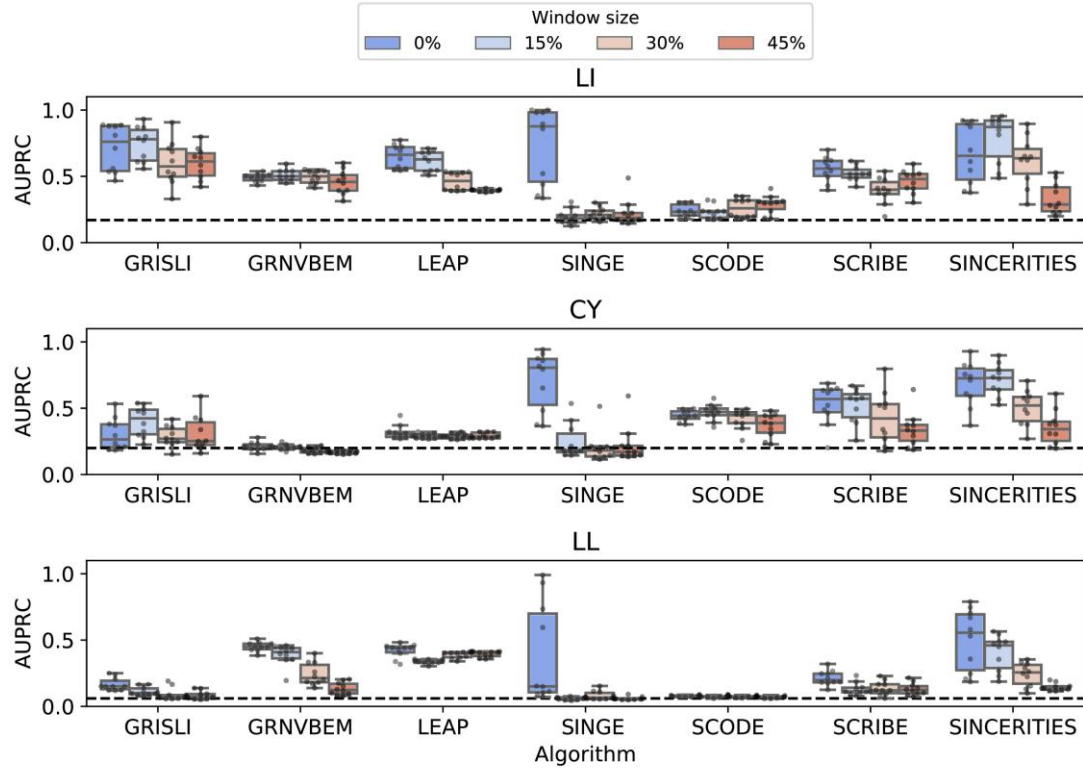


Supplementary Figure 10

Summary of AUPRC ratio values for experimental single-cell RNA-Seq datasets with 500 and 1000 genes.

Summary of AUPRC ratio values for experimental single-cell RNA-seq datasets. The left half of the figure (500 genes) shows results for datasets composed of the 500 most-varying genes. Each row corresponds to one scRNA-seq dataset. The first three columns report network statistics. The next six columns report AUPRC ratios. The right half (1000 genes) shows results for the 1000 most-varying genes. In both sections, algorithms are sorted by median AUPRC ratios across the datasets (rows) for the 500 gene set. For each dataset, the color in each cell is proportional to the corresponding value scaled between 0 and 1 (ignoring values that are less than that of a random predictor, which are shown as black squares). We display the highest and lowest values for each dataset inside the corresponding cells. Abbreviations: GENI: GENIE3, GRNB: GRNBoost2, PCOR: PPCOR, SINC: SINCERITIES.

(a)



(b)

Network	Window size	Correlation
LI	15%	0.99
	30%	0.96
	45%	0.90
CY	15%	0.99
	30%	0.96
	45%	0.90
LL	15%	0.99
	30%	0.96
	45%	0.90

Supplementary Figure 11**Effect of pseudotime shuffling on AUPRC values for synthetic networks.**

(a) Comparison of AUPRC values for networks inferred with original pseudotimes (blue) and shuffled within three window sizes 15% (light blue), 30% (beige) or 45% (orange). Each row corresponds to one of the three synthetic networks with single trajectories (Linear, Cycle, and Linear Long). Each boxplot represents 10 AUPRC values (10 datasets each containing 2,000 cells). The gray dotted line indicates the AUPRC value for a random predictor (i.e., the network density). In each boxplot, the box shows the 1st and 3rd quartile, and whiskers denote 1.5 times the interquartile range. (b) Median Pearson correlation between original and shuffled pseudotimes. For each network and window-size the median is over 10 datasets.

Supplementary Note: Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data

Aditya Pratapa¹, Amogh P. Jaliha², Jeffrey N. Law², Aditya Bharadwaj¹, and T. M. Murali¹

¹Department of Computer Science, Virginia Tech, Blacksburg, USA 24060

²Genetics, Bioinformatics, and Computational Biology Ph.D. Program, Virginia Tech, Blacksburg, USA 24060

Contents

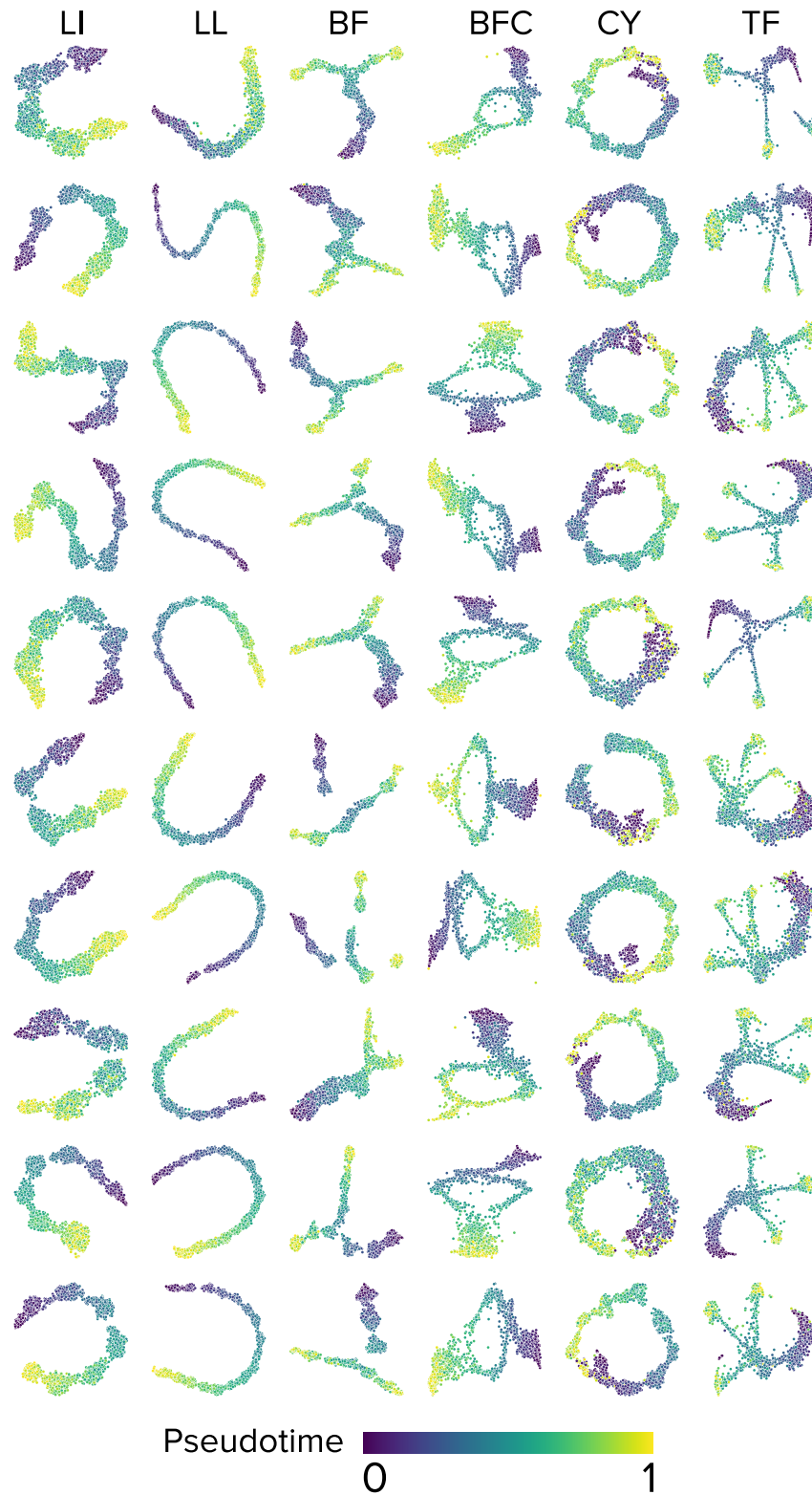
1	Datasets from Synthetic Networks	1
1.1	Generation of simulated datasets	1
1.2	Parameter Search	3
1.3	Effect of number of cells on AUPRC	5
1.4	Fidelity of simulations from inferred GRNs	6
1.5	Effect of using trajectory information	7
2	Datasets from Curated Models	9
2.1	Correspondence of BoolODE simulations and steady states of Boolean models	9
2.2	Parameter Search	14
2.3	Effect of dropouts on AUPRC	14
2.4	Comparing similarities of algorithm outputs	15
2.5	Network motifs	17
2.6	Explaining false positives in top- k networks	17
3	Experimental Single-Cell RNA-Seq Datasets	19
3.1	Parameter search	19
3.2	Correspondence of clusters in GRNs and gene expression datasets	21
3.3	Effect of gene selection strategy	22
3.4	Stability across multiple runs	23
3.5	Ensembles of GRNs	24
4	Software Details	26

1 Datasets from Synthetic Networks

1.1 Generation of simulated datasets

For each network, using the procedure outlined in “Creating Datasets from Synthetic Networks”(Methods), we applied BoolODE by sampling parameters ten times and creating 5,000 simulations per parameter set. Then we created two datasets per parameter set, one with 2,000 cells and another with 5,000 cells by sampling one cell per simulation, to obtain 20 different expression datasets. We visually inspected the two-dimensional representation of these datasets computed by t-SNE and confirmed that the simulation of each synthetic network yielded the expected trajectory. Next, we clustered the simulations in each dataset using k -means clustering with k set to the expected number of trajectories (Methods).

We plotted each cluster with a different colour in the two-dimensional t-SNE layouts. We visually compared the corresponding plots to confirm that each cluster indeed included cells from a distinct trajectory. Supplementary Figure 1(b) shows a t-SNE plot for an example 2,000-cell dataset for each network, with all these plots appearing in Supplementary Note Figure 1. Each colour in the example t-SNE plots in Supplementary Figure 1(c) corresponds to a distinct cluster. We found that the t-SNE visualizations succeed in capturing the qualitative steady states of each of the models we simulate. Specifically, the number of steady states we expected from each synthetic network (e.g., one steady state for linear, two for bifurcating, etc.) were visually apparent in the t-SNE plots. These results reassured us that BoolODE was successful in correctly simulating the network models.



Supplementary Note Figure 1: t-SNE visualizations of simulations of each artificial network with 10 distinct parameter sets and 2,000 cells. Each column corresponds to one synthetic network, indicated by the abbreviation on the top. Each row corresponds to one of the ten parameter sets. Each visualization is the 2-D t-SNE projection of the simulation output for the corresponding network and parameter set. The colors indicate the simulation time (blue for low, green for intermediate, and yellow for high). Abbreviations: LI: Linear, CY: Cycle, LL: Linear Long, BF: Bifurcating, BFC: Bifurcating Converging, TF: Trifurcating.

1.2 Parameter Search

Six of the 12 algorithms we tested had no parameters. For each of the other six methods, we performed a parameter sweep as described below. For the synthetic networks, we optimized the parameters on the datasets from the Linear network, separately for the five sets of 10 datasets containing varying number of cells (100, 200, 500, 2,000 and 5,000). For each method and number of cells, we chose the parameter value or combination of values that maximized the median AUPRC computed over the 10 datasets.

Below we describe the parameters tested for each method. We also provide brief comments on the effects of the parameters on the AUPRC.

LEAP. This method has a single parameter, `maxLag`, which controls the maximum amount of lag to consider. The authors of this method suggest not using a value larger than $1/3$. Hence, we tested seven values in the interval $[0, 0.33]$: 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, and 0.33.

SCODE. While SCODE has two parameters, `D`, and the number of iterations, we only varied the parameter `D`. For the number of iterations, we used 100, the same value as used by the authors. We also experimented with 1,000 iterations but found that this value did not improve the AUPRC (data not shown). In the documentation of the SCODE code, the authors suggest averaging the results from multiple runs. Therefore, we averaged the results across five runs. For `D`, we tested the values that they evaluated in their paper: 2, 4, 6, 8, and 10. We found this parameter had a large effect on the AUPRC and thus for each dataset, we used the value that gave the highest AUPRC.

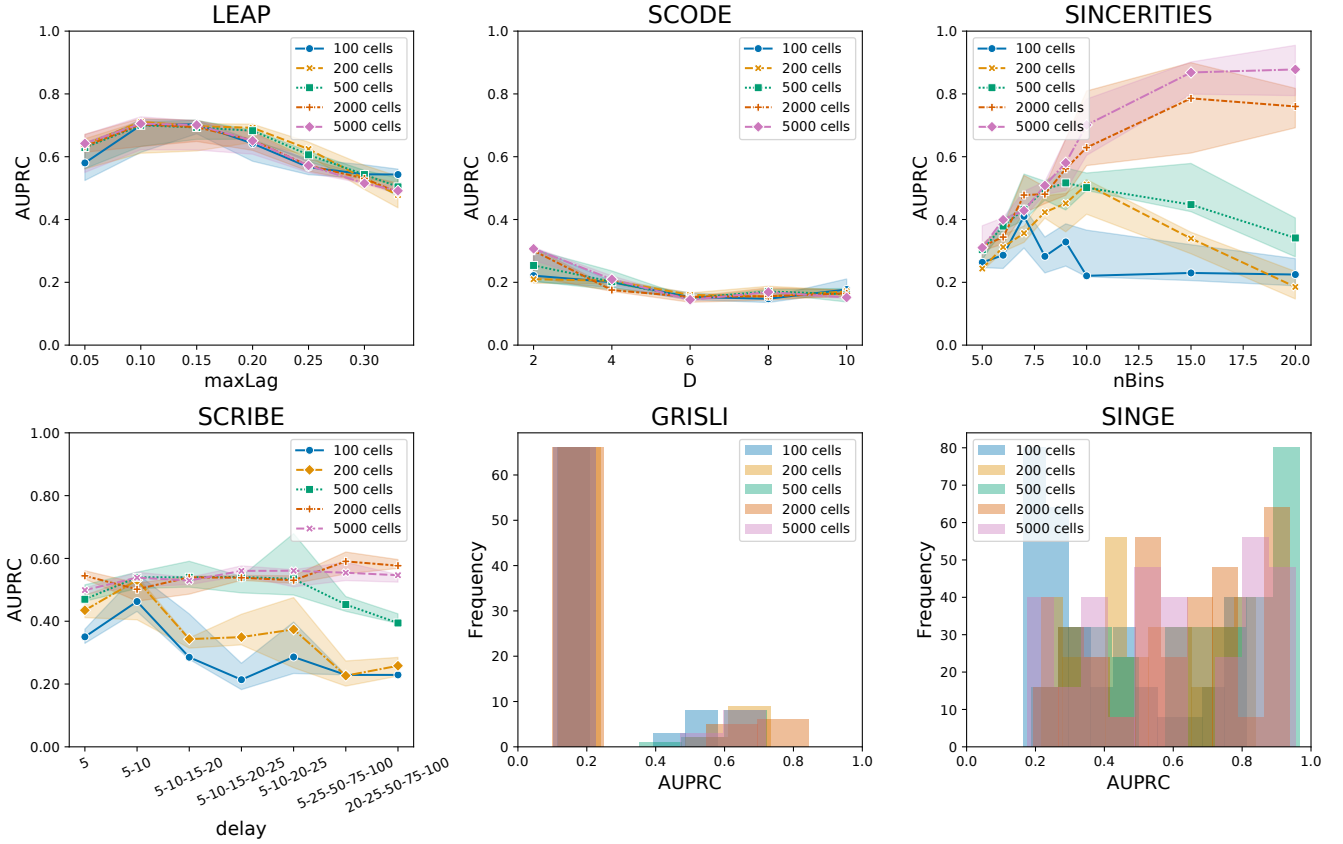
SINCERITIES. SINCERITIES does not need any parameters when using temporal transcriptional data. However, when experimental time points are not available, the authors suggest discretizing pseudotime values into bins. We varied the number of bins between 5, the smallest number of time points their code would accept, and 20 using the specific values of 5, 6, 7, 8, 9, 10, 15, and 20. We did not consider values larger than 20 reasoning that datasets with so many timepoints were unlikely to be available.

SCRIBE. The authors used the combination “5,10,20,25” for a parameter called “delay.” We evaluated this value as well as several other combinations of delays: “5”, “5,10”, “5,10,15,20”, “5,10,15,20,25”, “5,10,20,25”, “5,10,25,50,75,100”, and “20,25,50,75,100”.

GRISLI. This method has three parameters: `R`, `L`, and α . The authors suggest that `R` be as large as possible to reduce random fluctuations of the algorithm. We chose to use `R`= 3,000. Following the authors’ recommendation to test `L` and α over a large grid, we tried values in the same ranges as in their paper: `L` = 1, 5, 10, 25, 50, 75, and 100 and α = 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1.0. In the case of datasets from curated models, we varied α from 0 to 1.0 in steps of 0.25 in order to reduce the number of combinations we had to test.

SINGE. SINGE has a total of seven parameters. For five of these parameters, we used the same values as the authors of SINGE did: δt and `L` combinations of (3,5), (5,9), (9,5), (5,15), (15,5); kernel width σ = 0.5, 1, 2, 4; probability of zero removal and probability of removing samples values = 0, and 0.2 each. We used a reduced set of values for the sparsity parameter λ : we tested λ = 0 or 0.01 and the number of replicates = 2 or 6. We chose these parameter values since our initial, exploratory analysis showed that increasing λ to the other values used in SINGE (0.02, 0.05, 1) often resulted in decreased performance or very few or no output edges. We did not see any improvement in increasing the number of replicates to 10 either. These parameter values resulted in a total of 320 combinations. While the authors of SINGE used an ensemble of parameter values, we chose to simply use the combination of parameters that resulted in the best AUPRC. Since varying the parameters corresponding to the number of replicates, probability of zero removal and the probability of removing samples did not affect the AUPRC, we chose 2, 0 and 0 for their values. For λ , small values of 0 or 0.01 were best. δt , `L` and σ had the largest effect on AUPRC.

Supplementary Note Figure 2 summarizes the results from the parameter sweep and Supplementary Note Table 1 displays the best parameter value(s) for each method. We used these parameter values for the corresponding number of cells for the five other synthetic networks.



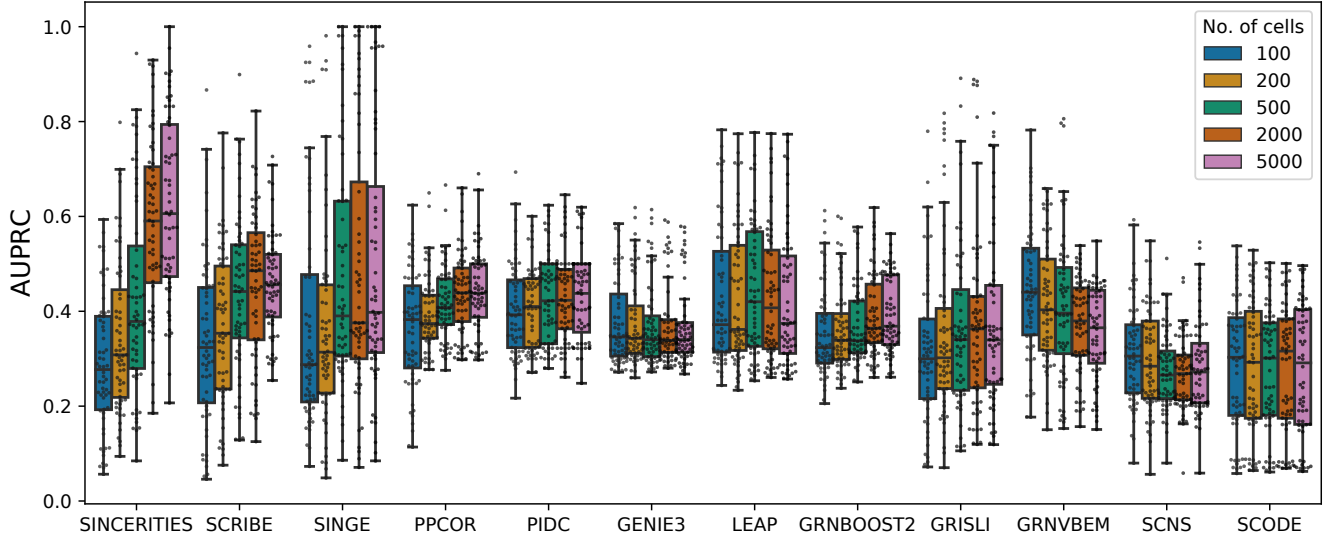
Supplementary Note Figure 2: Results of parameter search for each of the six methods with one or more parameters on the datasets (10 each) with 100, 200, 500, 2,000 and 5,000 cells for the Linear synthetic network. For the four methods with one parameter to vary, SCRIBE, SCODE, SINCERITIES, and LEAP, we show the median AUPRC (center line) as well as the 68% confidence interval of the median (shaded region), estimated using 1,000 bootstrapped samples of the data. For GRISLI and SINGE, which have two and seven parameters, respectively, we show histograms of the AUPRC over all parameter combinations we tested.

No. cells	LEAP maxLag	SCODE D	SINCERITIES nBins	SCRIBE delay	GRISLI L α	SINGE λ δt L σ
100	0.15	2	7	5,10	5 0.5	0.01 3 5 2
200	0.1	2	10	5,10	5 0.5	0 15 5 0.5
500	0.1	2	9	5,10,15,20,25	5 0	0.01 5 15 0.5
2000	0.1	2	15	5,25,50,75,100	5 0	0 9 5 0.5
5000	0.1	2	20	5,10,20,25	5 0	0 9 5 0.5

Supplementary Note Table 1: Parameter values that resulted in the highest median AUPRC for a given number of cells on the Linear synthetic network.

1.3 Effect of number of cells on AUPRC

To examine the effect of the number of cells on accuracy, for each algorithm, we compared the AUPRC values across all datasets from synthetic networks for 100, 200, 500, and 2,000 cells to those for 5,000 cells (Supplementary Note Figure 3). No algorithm had a statistically significant result for the 2000–5000 cell comparison (Supplementary Note Table 2). Four algorithms (GRNBoost2, PIDC, PPCOR, SINCERITIES) had significantly lower AUPRC values for 500 cells in comparison to 5000 cells. In addition to these four methods, SINGE and SCRIBE were significant in the 200–5000 cell comparison, with these two approaches and GRISLI also being significant in the 100–5000 cell comparison. The number of cells had no significant effect on five algorithms (GENIE3, GRNVBEM, LEAP, SCNS, and SCODE).



Supplementary Note Figure 3: Comparison of AUPRC values from all datasets from synthetic networks ($n = 60$), for each number of cells. In every boxplot, lower and upper hinges denote the 1st and 3rd quartiles, with whiskers extending to the largest value less than 1.5 times the interquartile range.

Number of cells compared	SINCERITIES	SCRIBE	SINGE	PPCOR	PIDC	GENIE3
100–5000	9.18×10^{-10}	7.85×10^{-6}	2.91×10^{-4}	5.28×10^{-6}	8.18×10^{-4}	1.0
200–5000	3.07×10^{-9}	2.49×10^{-5}	2.71×10^{-5}	2.47×10^{-6}	2.62×10^{-5}	1.0
500–5000	6.86×10^{-8}	0.30	0.35	2.36×10^{-6}	2.93×10^{-2}	0.67
2000–5000	0.12	0.71	0.31	0.35	0.40	1.0
Number of cells compared	LEAP	GRNBoost2	GRISLI	GRNVBEM	SCNS	SCODE
100–5000	1.0	5.62×10^{-4}	2.26×10^{-3}	1.0	1.0	1.0
200–5000	1.0	2.02×10^{-4}	0.41	1.0	1.0	1.0
500–5000	1.0	5.71×10^{-4}	0.59	1.0	0.91	0.88
2000–5000	1.0	0.87	1.0	1.0	0.66	1.0

Supplementary Note Table 2: Table of Benjamini-Hochberg-corrected one-sided Wilcoxon signed-rank test q -values corresponding to Supplementary Note Figure 3. Bold text highlights q -values ≤ 0.05 .

1.4 Fidelity of simulations from inferred GRNs

We sought to determine if a GRN inferred by an algorithm from datasets from synthetic networks, when represented as a Boolean network, would exhibit dynamics and steady states identical to the original models. We first verified that the synthetic networks yielded the expected number of steady states under asynchronous updates, namely, zero for Cycle, one each for Linear, Linear Long, and Bifurcating Converging, two for Bifurcating, and three for Trifurcating network.

We removed all self-loops from the ranked edges lists output by each algorithm. Then, we traversed the ranked list of edges output by each algorithm until each gene had at least one incoming interaction. SCNS produced such a network only for the Linear Long dataset; we attribute this result to the fact that we use a uniform value of threshold parameter for all genes due to the high running time of the method. For the output from GRNVBEM, SCODE, and SINCERITIES, we used the signs of interactions (activation or repression) predicted by that algorithm. For every other algorithm, we assigned the signs of interactions based on the values computed by PPCOR. We added self-interactions if they were present in the ground truth network.

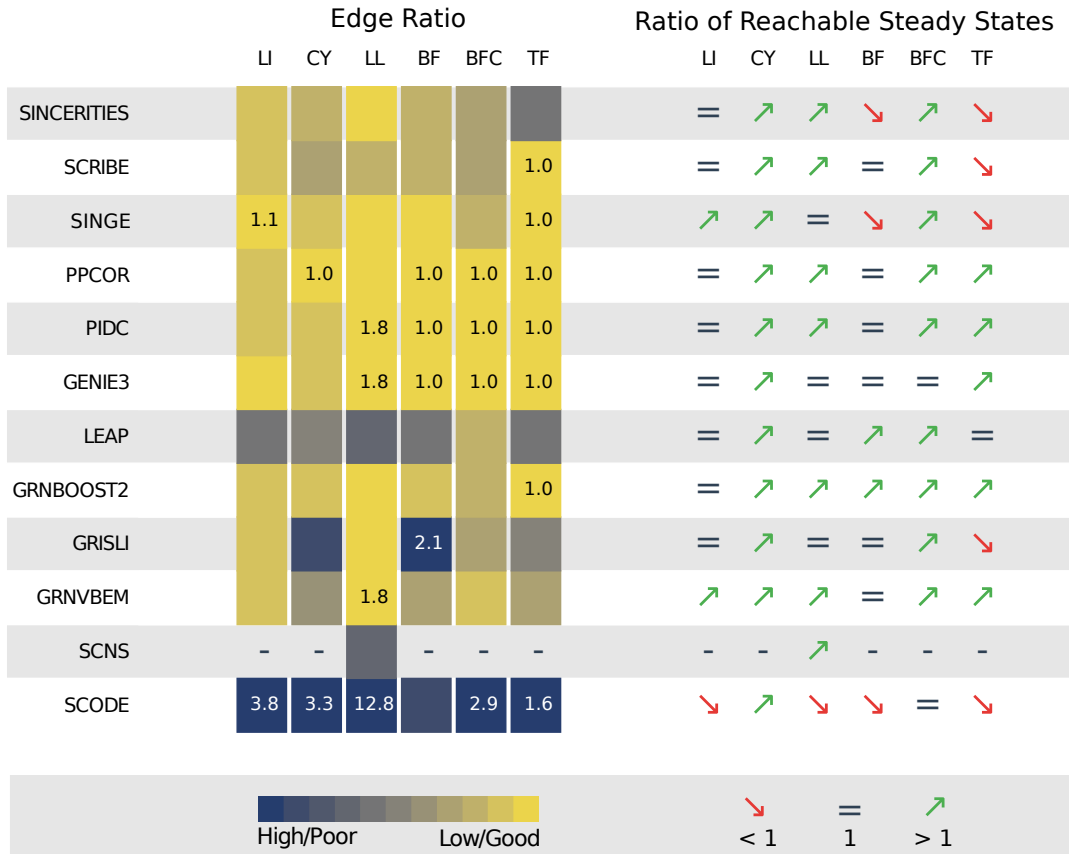
Next, we defined the rule for each gene as follows: a gene is ON if and only if at least one activator is ON and every inhibitor is OFF. We simulated this Boolean model asynchronously, i.e., at any state, we selected a gene uniformly at random and applied the rule for that gene to compute the next state. We computed the network formed by the transitions among all the possible states. With both the Boolean network and this “state space network” in hand, we computed two sets of statistics.

- (a) We recorded the number of the interactions in the Boolean network created from the computed GRN and its ratio with the number of interactions in the ground truth network.
- (b) We had simulated each ground truth network starting from a specific initial state, e.g., for the Linear model, we set g_1 to ON and all other genes to OFF. We asked how many steady states we could reach from this starting state in the Boolean network created from the computed GRN and recorded the ratio of this number with the number of steady states in the ground truth network.

Note that we did not perform this analysis for datasets from curated models since the Boolean rules in these networks were quite complex. We did not expect that the relatively simple rules used by BoolODE would suffice to recapture the properties of the Boolean networks we curated from the literature.

The six columns titled “Edge Ratio” in Supplementary Note Figure 4 present the number of edges in each such network divided by the number in the ground truth network. Almost all the methods required fewer than twice the number of edges in the ground truth networks, except for the Linear Long network. PPCOR achieved an edge ratio of 1.0 for four of six networks. SCODE had the largest edge ratios for all the networks.

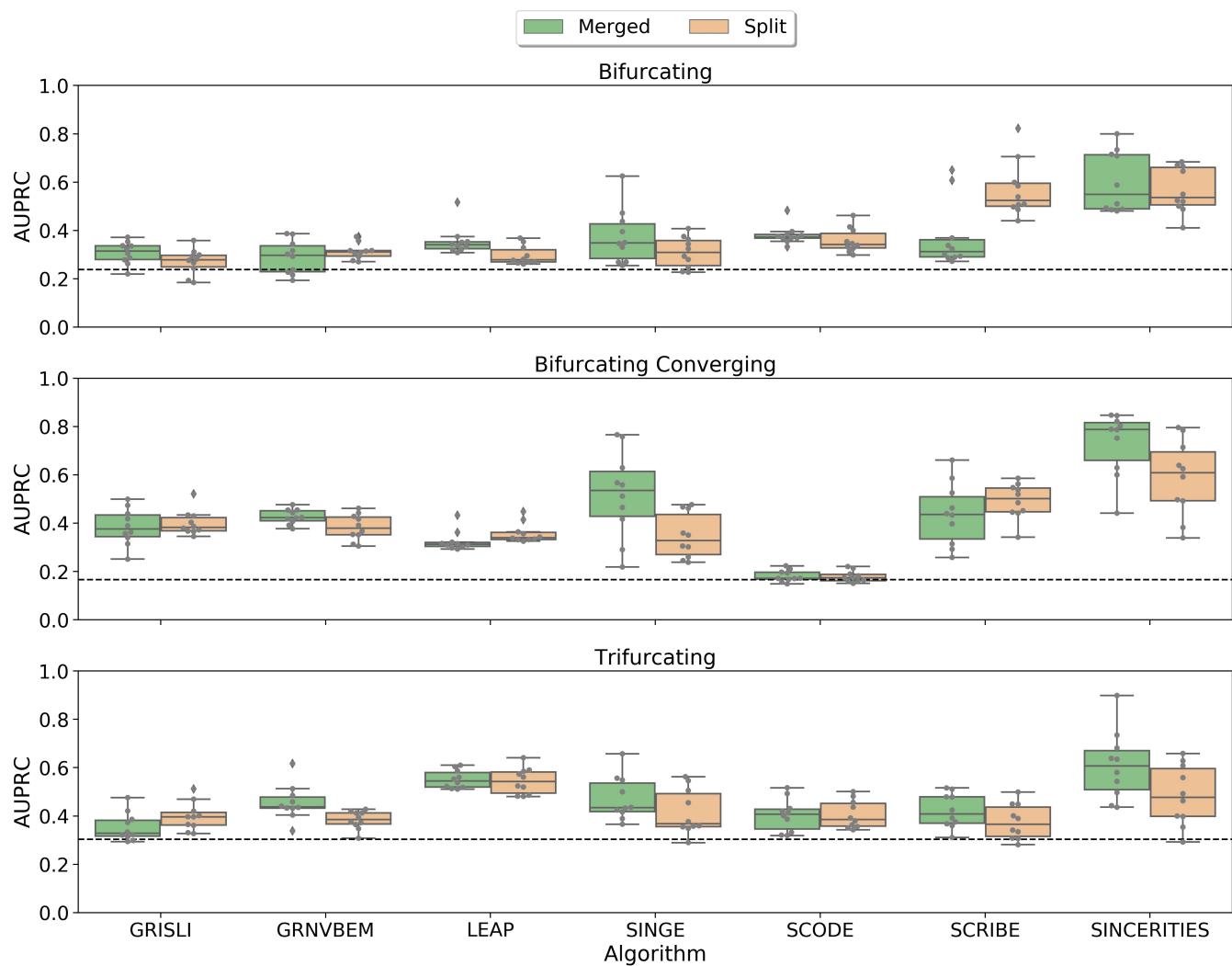
The columns titled “Ratio of Reachable States” in Supplementary Note Figure 4 display the number of steady states reachable from the initial condition used to simulate each ground truth network. Five of the twelve methods had a mean ratio of one for both the Linear and the Bifurcating networks, i.e., the number of steady states were one and two, respectively. While eight of the twelve methods yielded a mean of one steady state for the Linear network, only four methods had the same value for the Linear Long network. In general, most methods showed high edge ratios for the Linear Long network, they also over-predicted the number of steady states. We observed that all the methods predicted steady states for the Cycle network, failing to capture the expected limit cycle behavior. For the Bifurcating Converging network, the methods in general over-predicted the number of steady states, while for the Trifurcating network, all the methods (except LEAP) had either higher or lower median values for the number of reachable states.



Supplementary Note Figure 4: Summary of results for datasets from synthetic networks. The first set of six columns displays the median edge ratios. The next set of six columns display the median ratios of reachable steady states of the proposed GRN reconstructions for each of the six simulated datasets. The color is proportional to the corresponding value after scaling it between 0 and 1. For each dataset, we display the highest and lowest values inside the corresponding cells. We computed the medians over 20 simulated datasets for each synthetic network containing 2,000 and 5,000 cells. Abbreviations: LI: Linear, CY: Cycle, LL: Linear Long, BF: Bifurcating, BFC: Bifurcating Converging, TF: Trifurcating.

1.5 Effect of using trajectory information

Seven algorithms require the cells in the gene expression data to be ordered by (pseudo)time but cannot handle datasets representing multiple lineages. We compared the predictive performance of these algorithms in two situations: (a) split the cells into multiple trajectories, infer the GRN for each lineage, and combine these GRNs into network (see “Methods” for details), and (b) merge the trajectories into one and infer a single network. See “Datasets with multiple trajectories” in “Evaluation Pipeline” for details. We performed this comparison to check if access to information on a cell’s lineage gave these algorithms any advantage. By and large, there was virtually no difference between these two approaches for most algorithms (Supplementary Note Figure 5, Benjamini-Hochberg-corrected one-sided Wilcoxon signed-rank test q -value > 0.05). An exception was SCRIBE for which splitting trajectories yielded a superior AUPRC to the merged approach for at least two networks (q -value = 0.032). SINGE and SINCERITIES were contrasting exceptions, since merged trajectories performed better for all three networks (q -value = 0.007 and 0.004, respectively). This absence of a clear benefit of splitting over merging or vice versa may be a consequence of the numerous false positive interactions in the results, as indicated by the values of AUPRC.



Supplementary Note Figure 5: Comparison of AUPRC values with split (orange) and with merged (green) trajectories. Each row corresponds to one synthetic network that produces at least two trajectories. Each column corresponds to an algorithm. Each boxplot represents 10 AUPRC values (10 datasets each containing 2,000 cells). In every boxplot, lower and upper hinges denote the 1st and 3rd quartiles, with whiskers extending to the largest value less than 1.5 times the interquartile range.

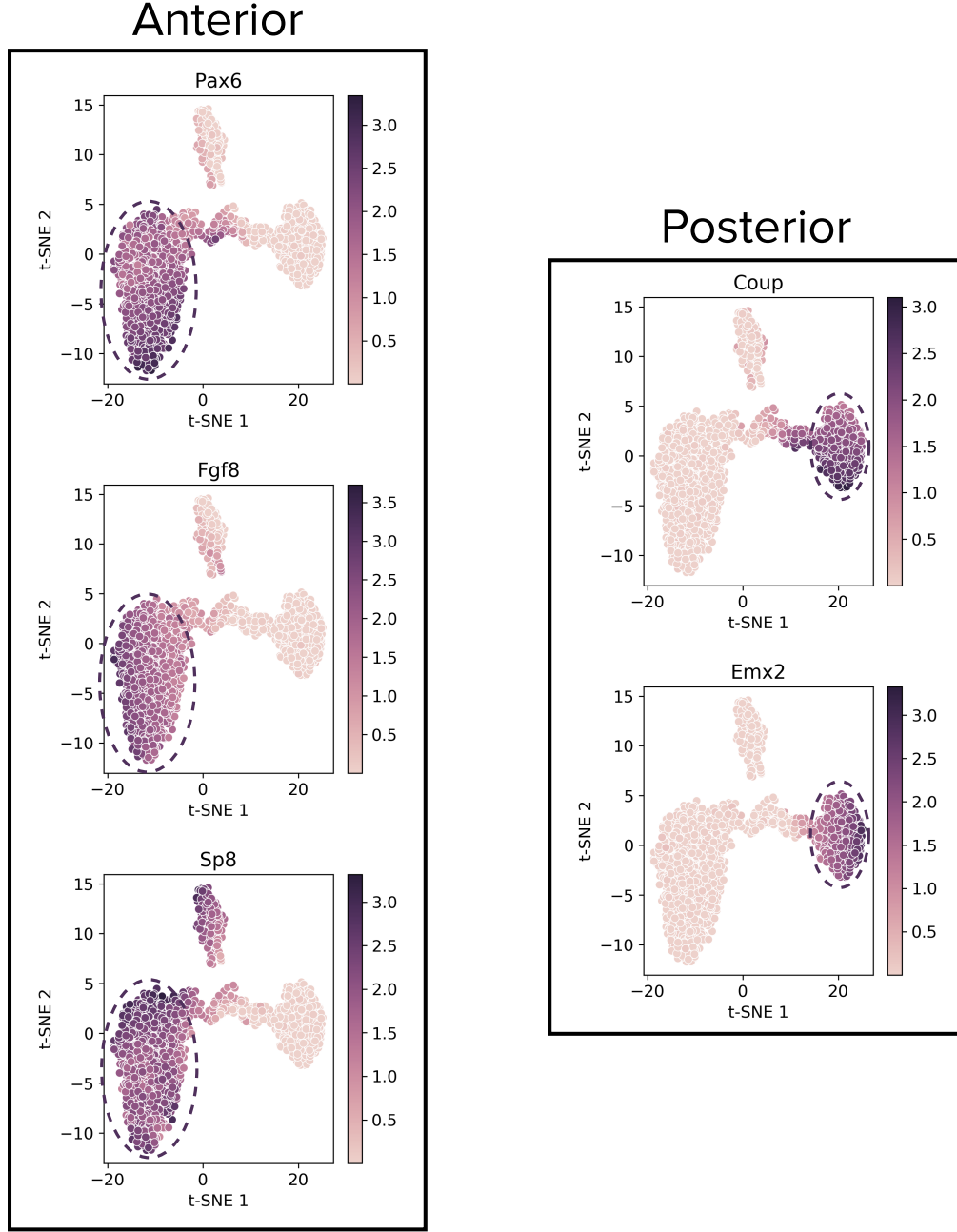
2 Datasets from Curated Models

2.1 Correspondence of BoolODE simulations and steady states of Boolean models

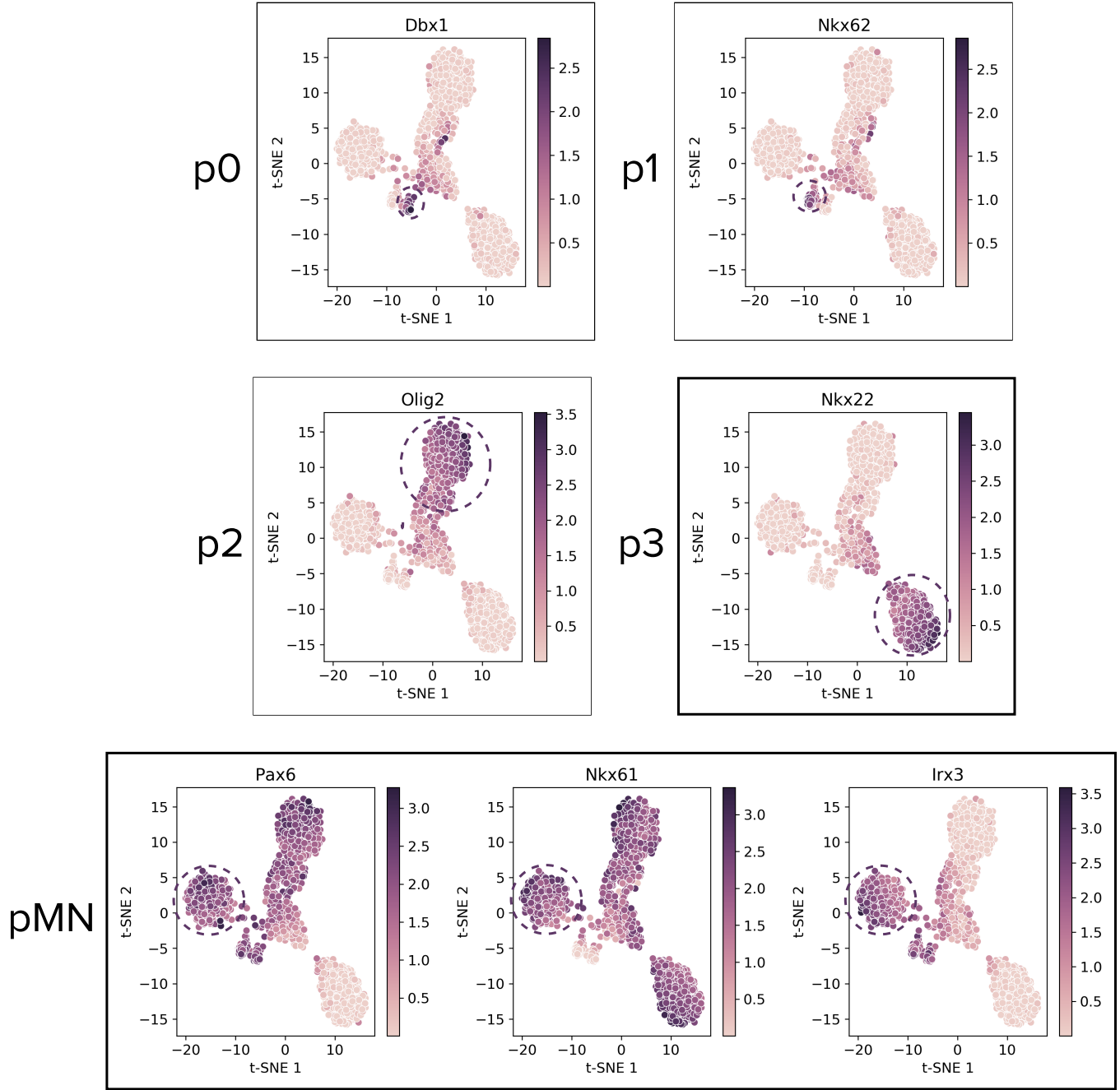
We sought to determine if the simulated datasets could capture steady state properties of the Boolean models, as reported in the corresponding publications. We performed this analysis in two stages.

(i) Figure 3(b) shows the two-dimensional t-SNE visualizations colored with simulation time for one of 10 datasets created by BoolODE for each Boolean model. Examining these plots visually, we observed continuous, discernible trajectories that ran along the entire simulation timeline, as seen by the colours of the cells changing gradually from blue to green to yellow. Next, we carried out k -means clustering of simulations with k set to the number of steady states of the model, as mentioned in the corresponding publication. Figure 3(c) displays these results, with a different color for each cluster (two for mCAD, five for VSC, four for HSC, and two for GSD). We visually corresponded the plots in Figures 3(b) and 3(c) to confirm that each cluster contained cells spanning the entire length of the simulation. After running the pseudotime inference algorithm Slingshot [1], we visualized the principal curves computed by this method in Figure 3(d); each such curve computed is a smooth representation of the trajectory. Therefore, we gave the pseudotime as input to the GRN algorithms.

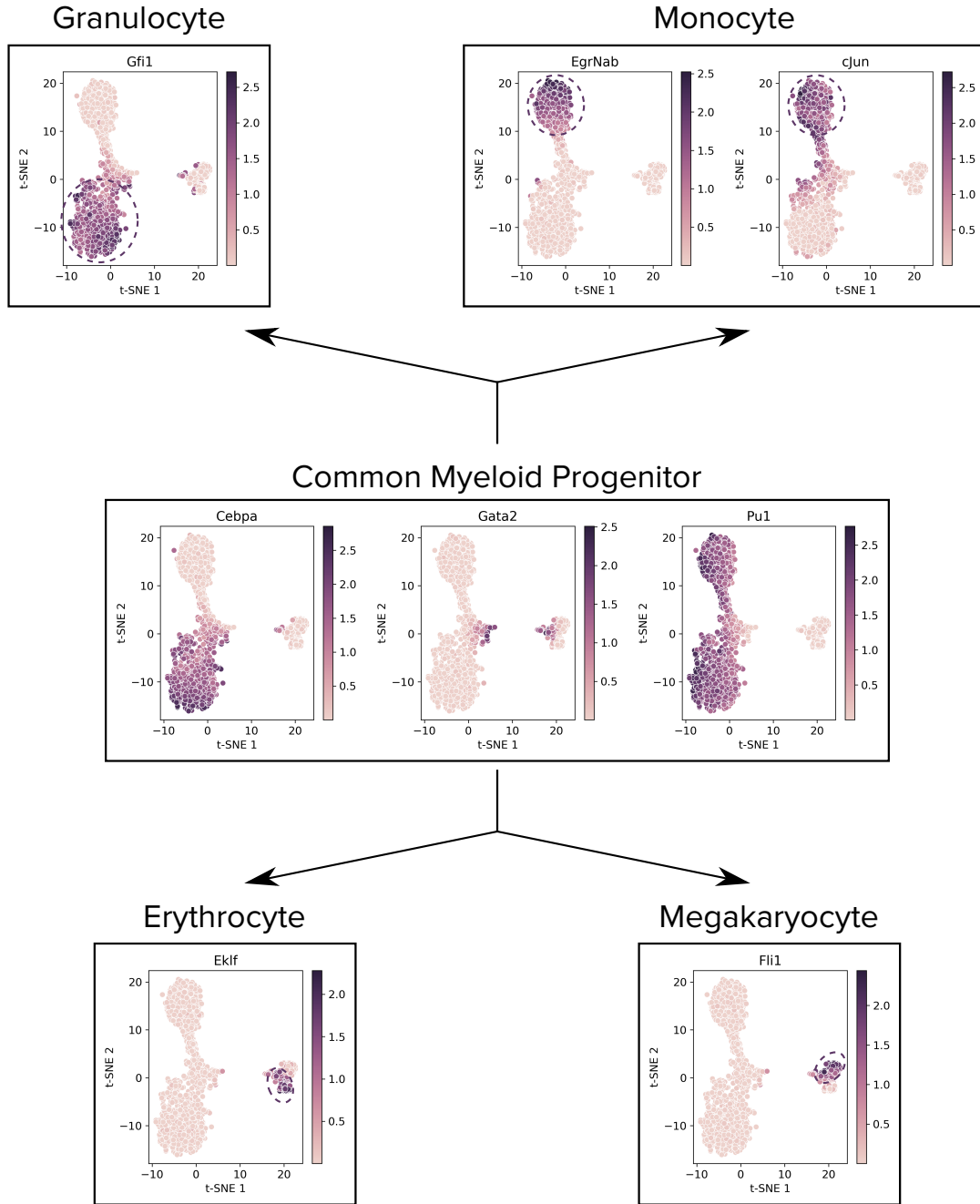
(ii) Each publication describing a Boolean model specifies a unique gene expression pattern that characterizes each steady state of that model. We asked if our simulated data matched these patterns. To this end, we colored each cell (a point on the 2D t-SNE plot) with the simulated expression value of each gene in that cell, with darker colors corresponding to larger values (Supplementary Note Figures 6 to 9). We explain the procedure we followed using an example. The mCAD model has two steady states: Anterior (characterized by up-regulation of Pax6, Fgf8, and Sp8) and Posterior (up-regulation of Coup and Emx2). In Supplementary Note Figure 6, we observed that Pax6, Fgf8, and Sp8 had high simulated expression values in a group of cells at the top right of the t-SNE plot whereas Coup and Emx2 had large values in the group in the bottom left. We concluded that these two cell groups corresponded to the Anterior and Posterior steady states of the Boolean model, respectively. We succeeded in finding similar matches for each of the other models (Supplementary Note Figures 7 to 9).



Supplementary Note Figure 6: t-SNE visualizations of simulated trajectories from the mCAD model. The bar on the right of each plot displays the mapping between the expression value of a gene and the corresponding color. The model has two steady states corresponding to the anterior and posterior compartments during cortical area development [2]. The anterior compartment is characterized by high transcriptional activity of Fgf8, Pax6 and Sp8, while the posterior compartment is characterized by the high transcriptional activity of Coup-tf1 and Emx2. The steady states corresponding to the two compartments appear as two distinct clusters in the t-SNE visualizations.

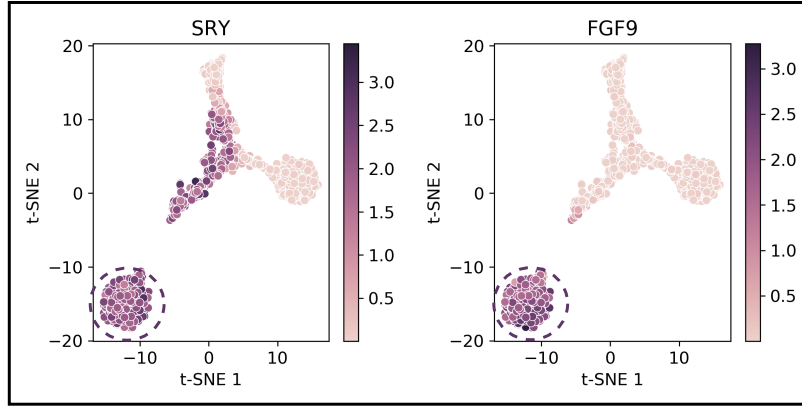


Supplementary Note Figure 7: t-SNE visualizations of simulated trajectories from the VSC model. This model exhibits five distinct steady states corresponding to five distinct cell types in the neural tube [3]. Each of these cell types appears as a distinct cluster in the tSNE visualizations shown above. The transcription factors specific to the p0 (Dbx1), p1 (Nkx6.2), p2 (Olig2), and p3 (Nkx2.2) states are shown in the four plots on top. The pMN state shown in the lower plot is characterized by high activity of Nkx6.1, Pax6 and Irx3.

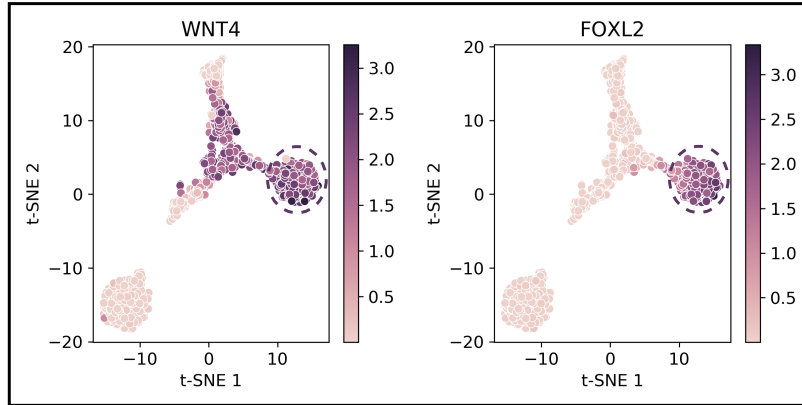


Supplementary Note Figure 8: t-SNE visualizations of simulated trajectories from the HSC model. In the initial state, which corresponds to the Common Myeloid Progenitor (CMP), the activities of GATA-2, PU.1 and C/EBP α are high [4]. There are two main branches of differentiation. The branch leading to the monocyte-granulocyte progenitors is characterized by the inactivation of GATA-2, followed by the activation of either Gfi1 (granulocyte) or cJun and EgrNab (monocyte) [4]. The clusters in the t-SNE visualization that correspond to these states are shown in the plots on top. Similarly, the branch leading to the megakaryocyte-erythrocyte progenitors is characterized by the inactivation of C/EBP α and PU.1 along with the concomitant activation of GATA1, FOG1 and SCL [4]. The subsequent inactivation of GATA2 along with activation of Fli1 leads to the megakaryocyte state while that with the activation of EKLF leads to the erythrocyte state [4]. These states are shown at the bottom.

Sertoli



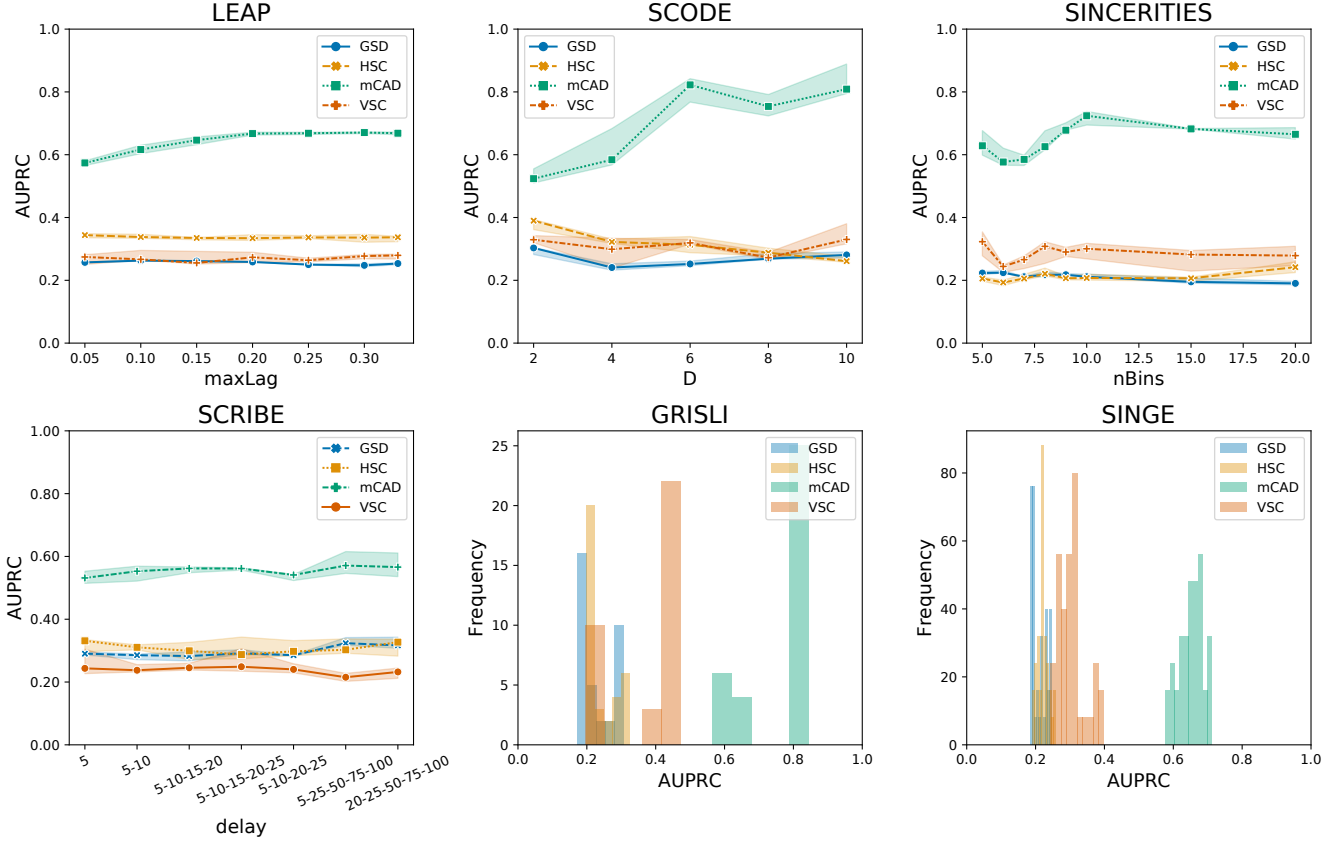
Granulosa



Supplementary Note Figure 9: t-SNE visualizations of simulated trajectories from the GSD model. This model has two steady states corresponding to Sertoli cells (male gonads) or Granulosa cells (female gonads) [5]. The clusters on the tSNE plots correspond to these two states. SRY and FGF9 show male-gonad-specific activity while WNT4 and FOXL2 show female gonad-specific activity. Except for GATA4 and WT1mKTS, which are active in both branches, all other transcription factors in the model belong to one of the two clusters shown above.

2.2 Parameter Search

For each of the six methods that require input parameters, we performed a parameter sweep using a procedure similar to the one described in Supplementary Note Section 1.2. However, there were a few notable differences. Since we only had datasets from curated models with a fixed number of 2,000 cells, we performed parameter sweep once for each model on the 10 non-dropout samples. For those algorithms that additionally use pseudotime information, we ran each algorithm on each individual trajectory for each dataset, combined the ranked lists of edges, and then computed the AUPRC. We chose this approach since every GRN inference algorithm in our evaluation that had parameters and had been developed explicitly for single-cell gene expression data could take only one trajectory as input. Since each of these datasets had 10 replicates, we chose the parameter value or combination of values that maximized the median AUPRC for each method. Supplementary Note Figure 10 shows the results for parameter search for the four different curated models and Supplementary Note Table 3 displays the best parameter value(s) for each method.



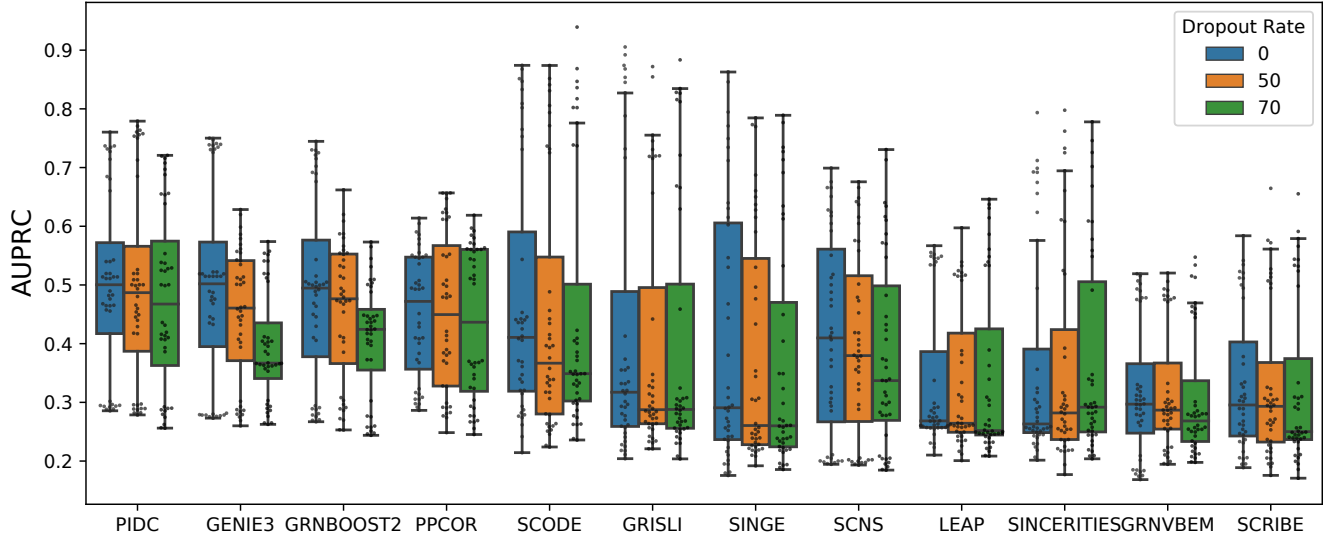
Supplementary Note Figure 10: Results of parameter search for each of the six methods with one or more parameters on the 10 datasets with 2,000 cells and no dropouts for the four curated Boolean models. For the four methods with one parameter to vary, SCRIBE, SCODE, SINCERITIES, and LEAP, we show the median AUPRC (center line) as well as the 68% confidence interval of the median (shaded region), estimated using 1,000 bootstrapped samples of the data. For GRISLI and SINGE, which have two and seven parameters, respectively, we show histograms of the AUPRC over all parameter combinations we tested.

2.3 Effect of dropouts on AUPRC

To study the effect of dropouts, for each algorithm, we compared the distributions of AUPRC scores across all datasets from curated models between the dropout rates $q = 0$ and $q = 50$ and between $q = 0$ and $q = 70$ (Supplementary Note Figure 11). Four GRN inference methods had a statistically significant difference in AUPRC values when comparing $q = 0$ to $q = 50$ (GENIE3, GRNBoost2, SCODE, GRISLI, Supplementary Note Table 4). For the other comparison, four additional algorithms were significant (PIDC, PPCOR, SINGE, SCNS), with SCODE being just above the 0.05 threshold. The four algorithms that were unaffected by dropout rates (LEAP, SINCERITIES, GRNVBEM, SCRIBE) had worse-than-random AUPRC values on the mCAD and VSC datasets.

Model	LEAP maxLag	SCODE D	SINCERITIES nBins	SCRIBE delay	GRISLI L	α	SINGE λ	δt	L	σ
GSD	0.1	2	6	5,25,50,75,100	10	0	0.01	15	5	0.5
HSC	0.05	2	20	5	10	0.25	0.01	3	5	1
mCAD	0.3	6	10	5,25,50,75,100	100	0	0	5	9	0.5
VSC	0.33	10	5	5,10,15,20,25	10	0	0.01	15	5	0.5

Supplementary Note Table 3: Parameter values that resulted in the highest median AUPRC for each curated model.



Supplementary Note Figure 11: Comparison of AUPRC values from the four datasets from curated models ($n = 40$), for three dropout rates. In every boxplot, lower and upper hinges denote the first and third quartiles, with whiskers extending to the largest value less than 1.5 times the interquartile range.

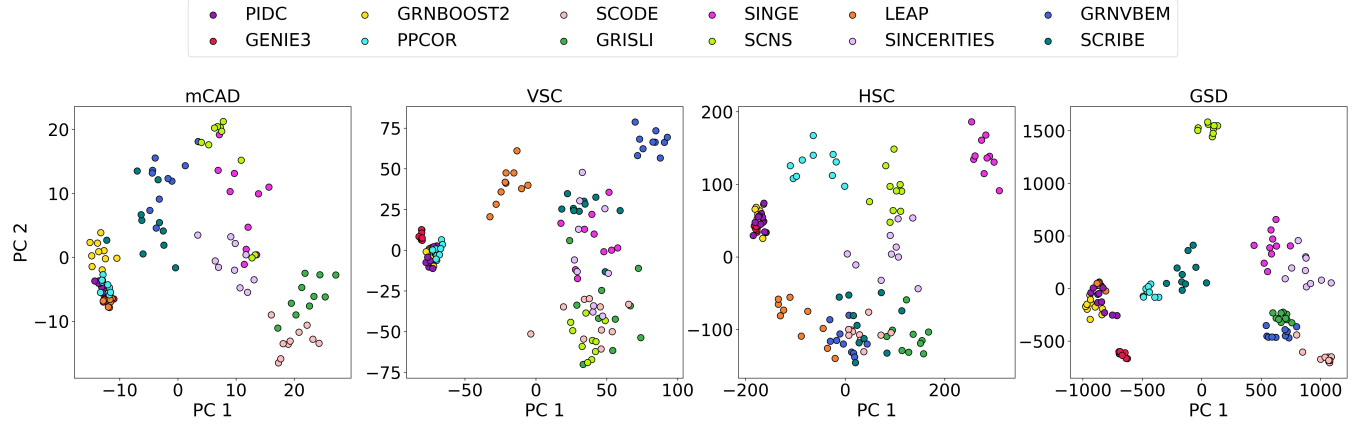
Dropout rates compared	PIDC	GENIE3	GRNBoost2	PPCOR	SCODE	GRISLI
0-50	0.15	4.54×10^{-3}	0.03	0.6	0.034	0.03
0-70	4.17×10^{-4}	5.78×10^{-3}	6.87×10^{-4}	0.03	0.051	0.02
Dropout rates compared	SINGE	SCNS	LEAP	SINCERITIES	GRNVBEM	SCRIBE
0-50	0.09	0.85	0.25	0.74	0.71	0.6
0-70	0.04	0.03	0.65	0.82	0.09	0.23

Supplementary Note Table 4: Benjamini-Hochberg-corrected q -values for the one-sided Wilcoxon signed-rank test corresponding to Supplementary Note Figure 11. Bold text highlights q -values ≤ 0.05 .

2.4 Comparing similarities of algorithm outputs

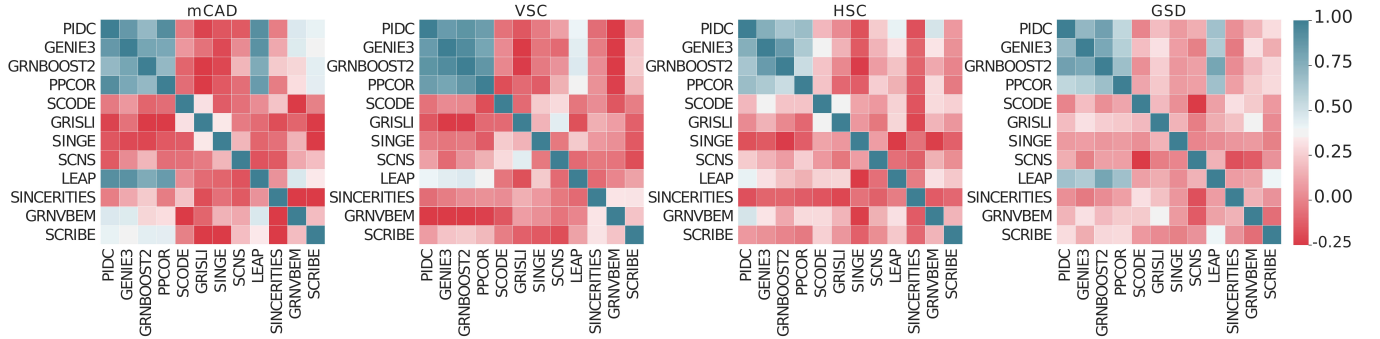
Our results indicate moderate to poor accuracy of the networks inferred by the 12 methods on datasets from curated models (Figure 4). We were curious if the algorithms made similar errors or if there were some subnetworks of the Boolean models that were easy for most algorithms to infer correctly. We performed three analyses for each model separately. (i) We converted each algorithm’s output into a vector of edges and recorded the edge ranks in this vector. We used PCA on the set of vectors on all 10 datasets and all 12 algorithms to compute a two-dimensional projection (Supplementary Note Figure 12). (ii) For each dataset we computed the Spearman’s correlation coefficients between these vectors for all pairs of algorithms. Supplementary Note Figure 13a presents the results for one representative dataset. (iii) For the top- k edges for each dataset, we computed the Jaccard indices between all pairs of algorithms. Supplementary Note Figure 13b presents the results for one representative dataset. The PCA results

suggest that some algorithms have somewhat similar outputs (e.g., GENIE3, GRNBoost2, and PIDC for all models, with LEAP and PPCOR each being similar to the first three methods for two of the four models). The Spearman's correlations and Jaccard indices echo these trends. GENIE3 and GRNBoost2 are both based on random forests, which may explain their similarities. Similarly, both LEAP and PPCOR are based on computing correlations. What is striking is that most pairs of Spearman's correlations are somewhat negative (between -0.25 and 0) and most pairs of Jaccard indices are less than 0.5, suggesting that the outputs are quite different from each other.

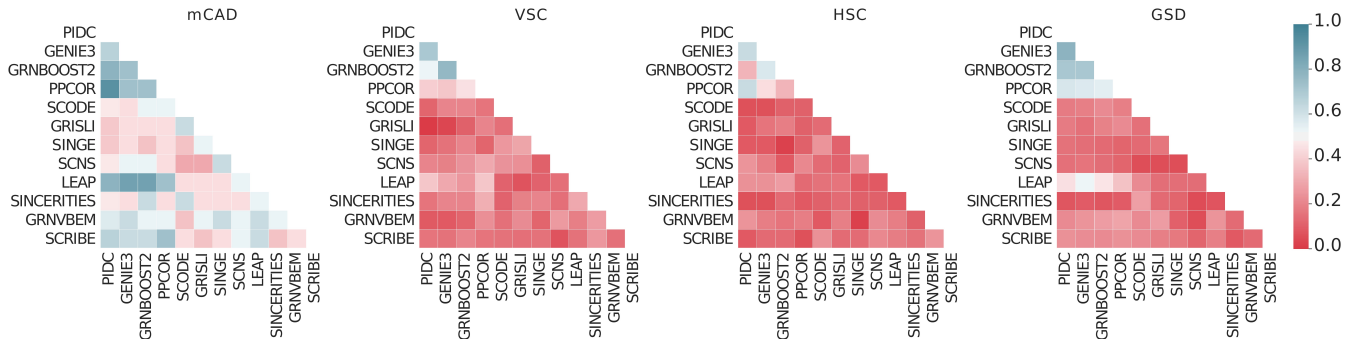


Supplementary Note Figure 12: PCA projections of edge rank vectors. Each point corresponds to a vector of edge ranks predicted by an algorithm for a dataset.

(a) Spearman correlation



(b) Jaccard index



Supplementary Note Figure 13: (a) Heatmap of the Spearman correlation matrix for a representative dataset. Each cell represents the correlation value between the edge rank vectors for a pair of algorithms. (b) Heatmap of the Jaccard index for a representative dataset. Each cell in the heatmap represents the Jaccard index between the sets of top- k edges for a pair of algorithms.

2.5 Network motifs

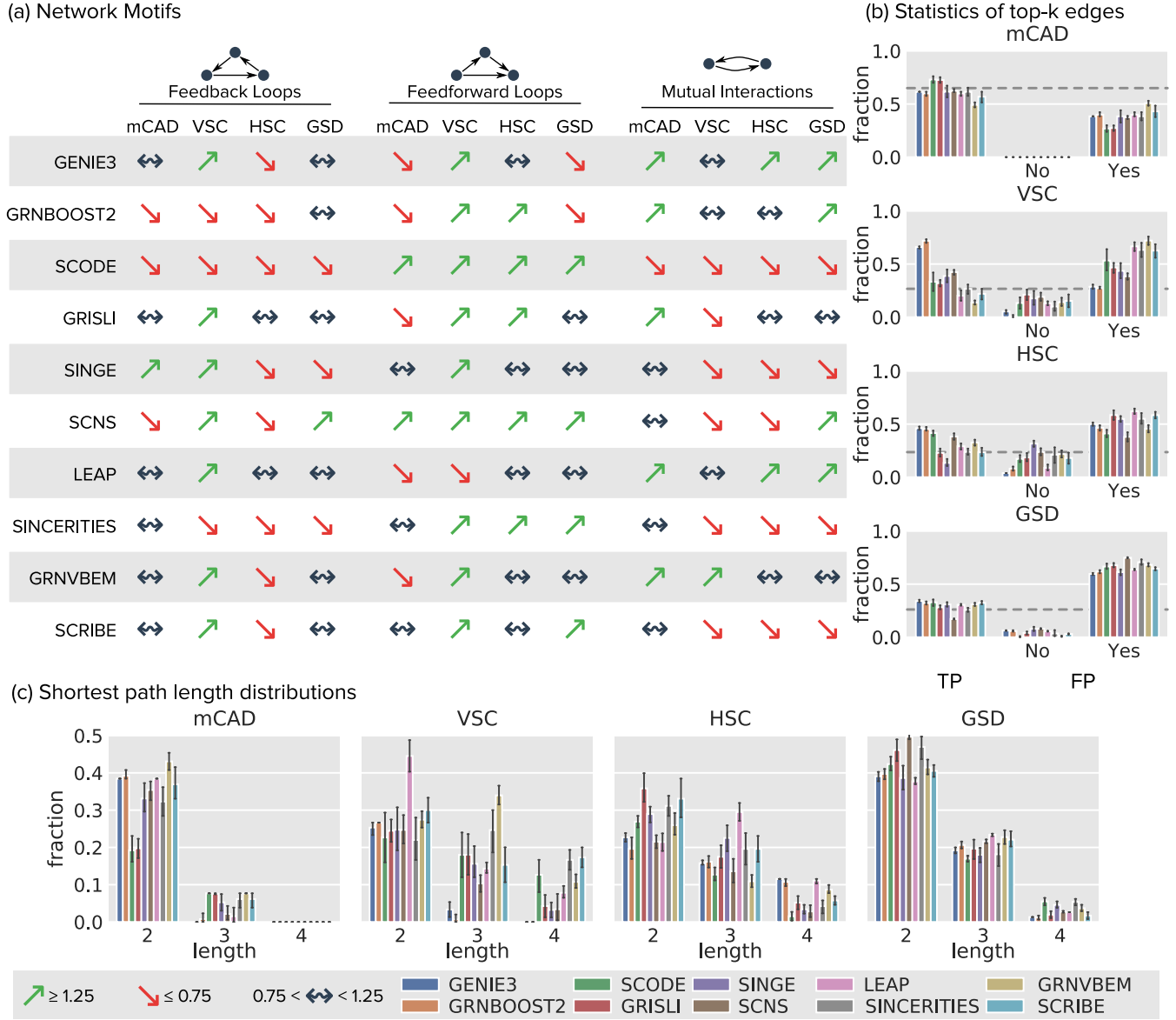
We investigated if certain network motifs were over-represented in the top- k networks. Specifically, we counted the number of three-node feedback and feedforward loops and (two-node) mutual interactions in the top- k networks from each GRN algorithm and reported their ratios with their respective values in the ground truth network. For this analysis, we ignored algorithms that predict only undirected edges (PIDC and PPCOR). We report the medians of these values (over the 10 datasets we simulated for each model) in Supplementary Note Figure 14(a). For the mCAD model, we observed that every GRN inference method yielded close to expected or lower than expected numbers of feedback and feed-forward loops compared to the ground truth network. This trend is likely due to the fact that the mCAD network is fairly dense. For the VSC model, which primarily has mutual inhibitory edges, 9 out of the 10 models had expected or lower than expected number of mutual interactions. For the HSC and GSD networks, we noted an overall decrease in feedback loops and mutual interactions but an increase in feedforward loops. In the case of feedback loops, we noted that LEAP and GRISLI were the only methods that predicted roughly as many or more feedback loops than in the ground truth network for each of the four Boolean models. SCODE and SINCERITIES typically produced fewer feedback loops and mutual interactions but more feedforward loops.

2.6 Explaining false positives in top- k networks

We reasoned that the poor value of early precision for a GRN inference algorithm could be due to its tendency to predict “indirect” interactions, i.e., if, in the Boolean model, gene a regulates gene b , which in turn regulates gene c , a method may also predict an interaction between a and c . This type of configuration is a feedforward loop, which we have seen to be in excess over the ground truth network for several algorithms (Supplementary Note Figure 14). To confirm the possibility of indirect interactions, we considered each false positive edge (u, v) in a top- k network and computed the length of the shortest path between u and v in the corresponding ground truth network. Supplementary Note Figure 14(b) and (c) summarize the results.

We discuss the results for the VSC, HSC, and GSD models first. We observed that the endpoints of about 5% to 20% of false positive edges were not even connected in the ground truth networks by any path (bars labeled “No” in Supplementary Note Figure 14(b)). Virtually all the remaining false positive edges corresponded to paths of lengths between two and four in the ground truth network (Supplementary Note Figure 14(c)); What was most striking was that the largest proportion of these false positive edges were connected by paths of length two, i.e., if added to the ground truth network, they would indeed form feed-forward loops. This result lends strong credence to our hypothesis that GRN inference algorithms are predicting indirect interactions.

In the case of the mCAD model, a very large fraction of false positive edges corresponded to paths of length two in the corresponding ground truth network. We reasoned that the density of the mCAD network was the cause for this statistic despite the lower prevalence of feedforward loops in the predicted networks for the mCAD model.



Supplementary Note Figure 14: (a) Analysis of network motifs in the predicted top- k networks using datasets from curated models. We considered three motifs, namely, feedback loops, feedforward loops and mutual interactions. We counted the number of occurrences of each motif type in each predicted top- k network, as well as in the ground truth network, and we report the ratio of these counts. Green upward pointing arrows represent ratios greater than 1.25, red downward pointing arrows represent ratios less than 0.75, and the black arrows represent ratios between 0.75 and 1.25. (b) Statistics on the edges in the top- k network for each method divided into three parts: fraction of top- k edges that are true positives (TP), fraction of top- k edges that are false positives and whose endpoints are not connected by any path in the ground truth network (FP, No), and the fraction of top- k edges that are false positives and whose endpoints are connected by a path (of length two or more) in the ground truth network (FP, Yes). Each colored bar corresponds to one algorithm. The height of a bar represents the mean and the error bar the 95% confidence interval over the 10 datasets. (c) For every false positive edge in each predicted top- k network, we computed the length of the shortest paths in the ground truth network that connected the endpoints of this edge. Bar plots show the fractions of top- k edges that are false positive edges and whose endpoints have paths of lengths 2, 3, and 4 in the ground truth network in the top- k networks, colored by the method. The height of a bar represents the mean and the error bar the 95% confidence interval over the 10 datasets.

3 Experimental Single-Cell RNA-Seq Datasets

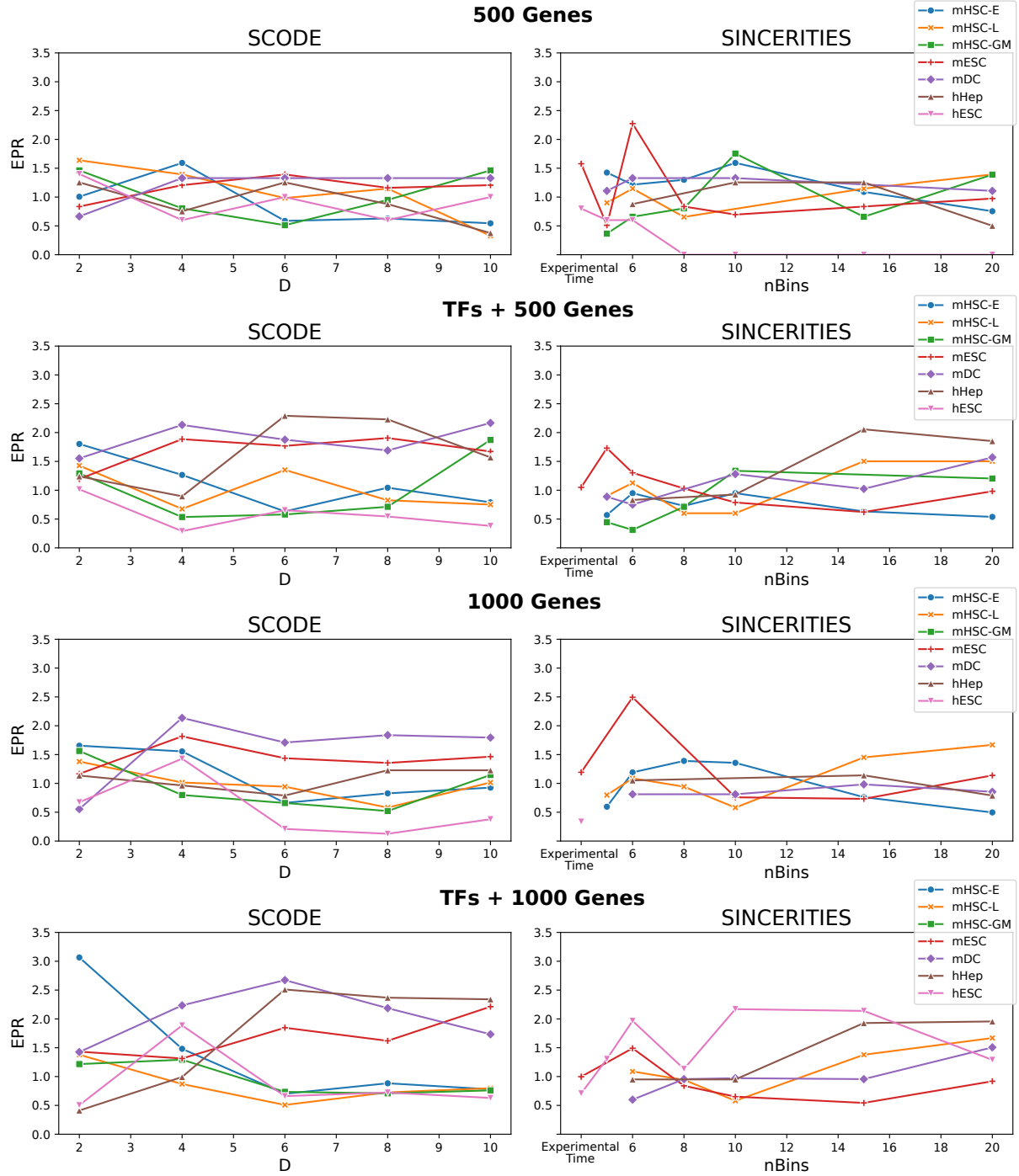
3.1 Parameter search

For the seven experimental single-cell RNA-Seq datasets and each of the four subsets of genes, we performed a parameter sweep for SCODE and SINCERITIES, and selected the parameter that yielded the highest EPR when evaluated on the non-specific ChIP-Seq ground truth network (Supplementary Note Figure 15, Supplementary Note Table 5). We used these parameter values for the other two types of ground truth networks.

In the case of SINCERITIES, which was developed for temporal single cell expression data, we also provided the experimental time as input for the mESC and hESC datasets. For five out of eight subsets of genes (across both datasets), SINCERITIES had a higher EPR when we first computed pseudotime and did parameter estimation than when we used experimental time (Supplementary Note Figure 15). In the case of the TFs+500 gene set for the hESC dataset, SINCERITIES gave a runtime error for experimental time and for every parameter value. For the 1,000 gene set in the same dataset, SINCERITIES ran successfully for experimental time but not for any parameter value. Only for the 500 gene set in the same dataset did the experimental time have a higher EPR than from parameter estimation.

Dataset	SCODE				SINCERITIES			
	D				nBins			
	500	TFs+500	1000	TFs+1000	500	TFs+500	1000	TFs+1000
mHSC-E	4	2	2	2	10	10	8	-
mHSC-L	2	2	2	2	20	15	20	20
mHSC-GM	10	10	2	4	10	10	-	-
mESC	6	8	4	10	6	5	6	6
mDC	10	10	4	6	10	20	15	20
hHEP	2	6	10	6	10	15	15	20
hESC	2	2	4	4	Exp. time	-	Exp. time	10

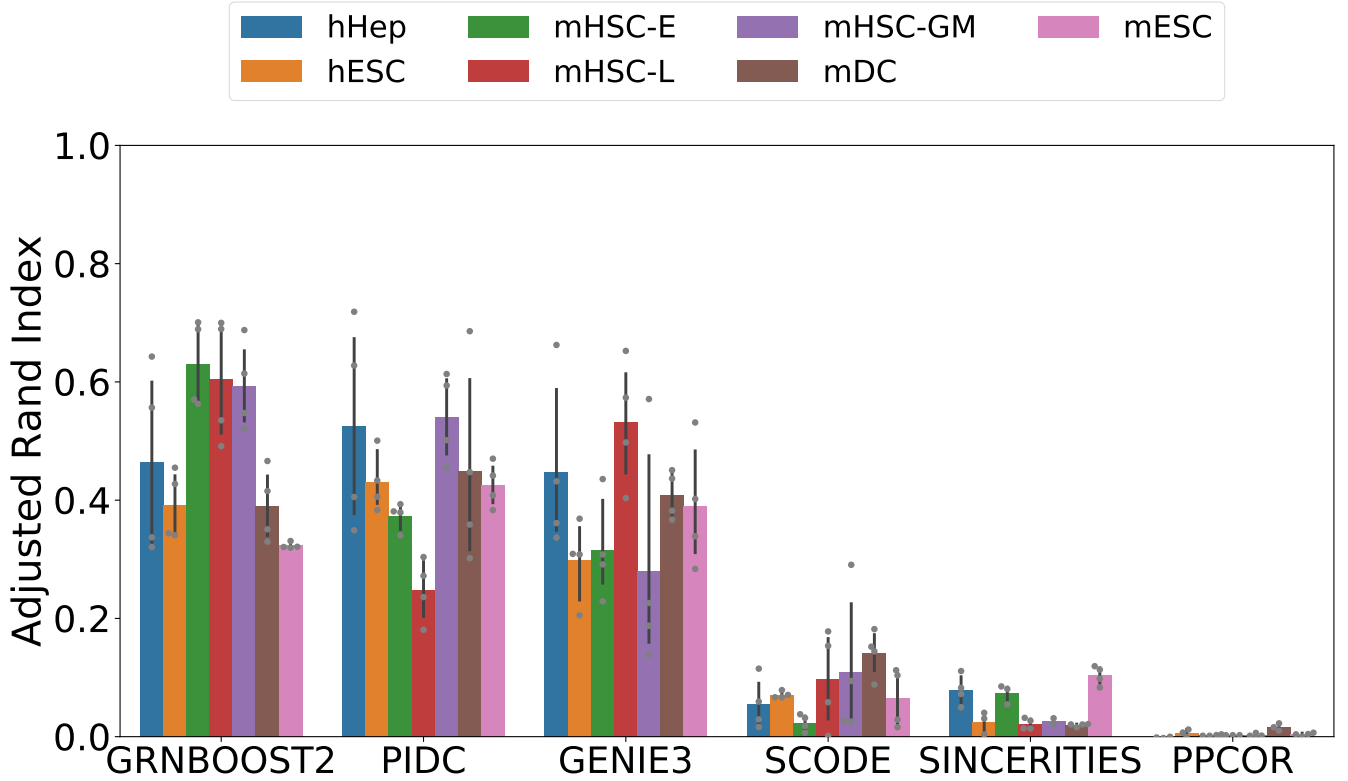
Supplementary Note Table 5: Parameter values that resulted in the highest EPR for a given experimental single-cell RNA-Seq datasets and gene set, evaluated on the non-specific ChIP-Seq networks.



Supplementary Note Figure 15: Results of the parameter search for SCODE and SINCERITIES on experimental single-cell RNA-Seq datasets, with four different sets of genes. We used the non-specific ChIP-Seq networks as the ground truth. For SINCERITIES, the point for “Experimental Time” corresponds to using the experimental time points available in the hESC and mESC datasets. Points that are missing correspond to cases where SINCERITIES gave an error and did not produce output.

3.2 Correspondence of clusters in GRNs and gene expression datasets

We investigated how much the clusters in the predicted GRNs agreed with those in the gene expression datasets. We first computed a co-expression network using the absolute value of the Pearson's correlation coefficient as the weight of the edge between two genes. Next, we ran the Louvain clustering algorithm [6] on the predicted GRN (considering all edges and their weights) and on the co-expression network to compute clusters. In order to compute clusters using the Louvain algorithm, we converted the GRNs into undirected networks: if a method predicted both the edge (a, b) and (b, a) , we used only the higher of the two edge weights. Next, we measured the similarity between these clusterings using the Adjusted Rand Index [7]. We recorded the values of this index for each pair of experimental single-cell RNA-Seq dataset and algorithm. Values of this index close to 1 indicate that the modules inferred from GRNs are similar to the ones detected using gene expression datasets. Averaged over all experimental single-cell RNA-Seq datasets, we observed that GRNBoost2 had the highest ARI value of 0.485 and PPCOR had the lowest ARI value of 0.004, i.e., close to random (Supplementary Note Figure 16). The ordering of the methods by average ARI was close to the ranking by median EPR, suggesting that modules in the more accurate GRNs have a good concordance with clusters computed directly from gene expression data.



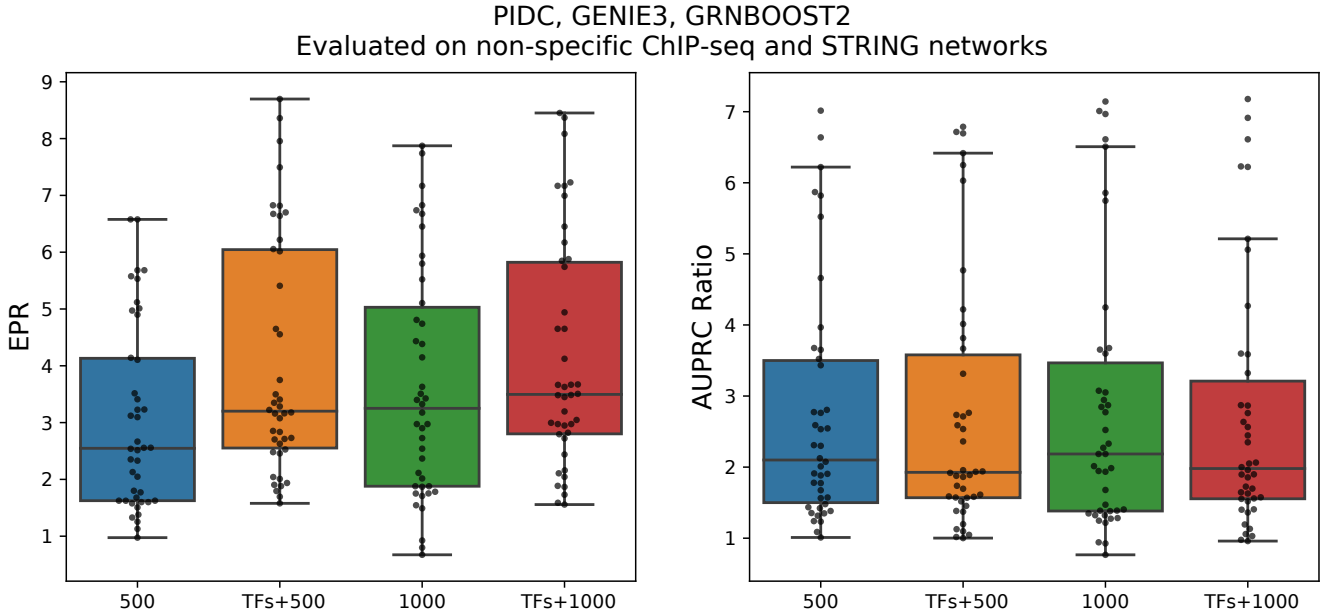
Supplementary Note Figure 16: Bar plot of Adjusted Rand Index values for clusterings generated using real gene expression datasets and GRNs predicted by an algorithm.

3.3 Effect of gene selection strategy

We sought to examine the effect of the number of genes (gene sets “500” and “1,000”) as well as including all TFs (sets “TFs+500” and “TFs+1,000”), on the performance of the methods. Therefore, for each gene set, we combined the EPR values of the three top performing methods (PIDC, GENIE3, and GRNBoost2) on the non-specific ChIP-Seq and STRING networks. A p -value of 0.017 for the Kruskal-Wallis test indicated that EPR values for these four sets of genes did not originate from the same distribution. To further investigate if these differences arose from the differences in the number of genes or the inclusion of TFs, we grouped the gene sets as follows:

- (i) To study the effect of addition of TFs, we combined EPR values for the gene sets “500” and “1000” and compared them to the union of the EPR values for “TFs+500” and “TFs+1000” ($n = 84$).
- (ii) To study the effect of the number of genes, we compared the union of the EPR values for “500” and “TFs+500” with the union of “1000” and “TFs+1000” ($n = 84$).

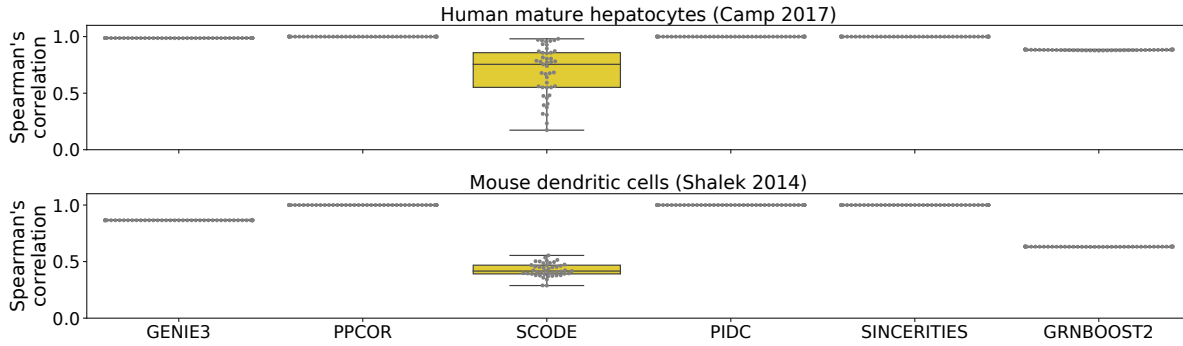
We found that including TFs resulted in a statistically significant improvement (one-sided Mann-Whitney U test p -value 0.003), while including more highly varying genes did not (p -value 0.084). Thus, our analysis suggests that including significantly varying TFs may be more beneficial than adding more highly-varying genes. The AUPRC ratio did not show statistically significant variation with the number of genes or the addition of TFs.



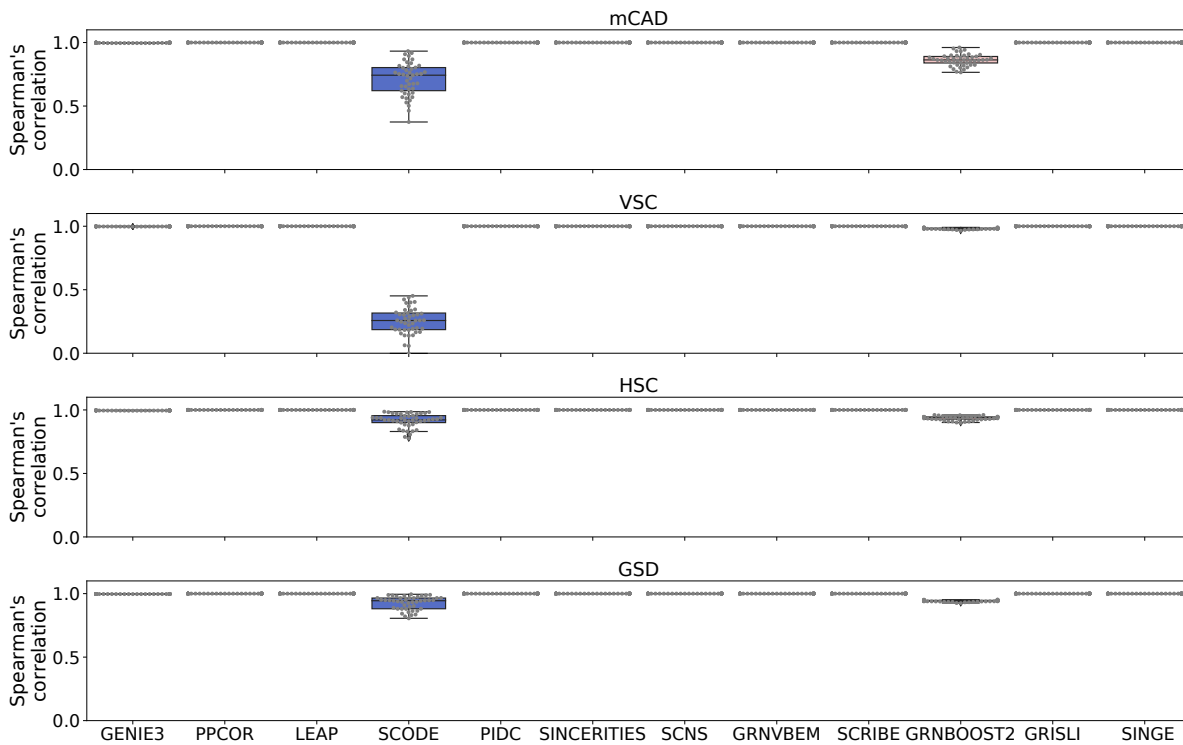
Supplementary Note Figure 17: Boxplots of EPR and AUPRC ratios of PIDC, GENIE3, and GRNBoost2 for experimental single-cell RNA-Seq datasets on the 500 most-varying genes, all significantly-varying TFs with the 500 most-varying genes, and similarly defined sets for 1,000 genes evaluated on the non-specific ChIP-Seq and STRING ground truth networks. Each box contains a total of 42 values. In each boxplot, lower and upper hinges denote the 1st and 3rd quartiles, with whiskers extending to the largest value less than 1.5 times the interquartile range.

3.4 Stability across multiple runs

(a) Experimental single-cell RNA-seq datasets



(b) Boolean networks



Supplementary Note Figure 18: Box plot of Spearman's correlations of edge ranks for 10 runs on (a) two representative experimental single-cell RNA-Seq datasets and (b) on four curated models. Each row corresponds to one dataset or model. Each column corresponds to an algorithm. Each box plot represents 45 Spearman's correlation values. In each boxplot, lower and upper hinges denote the 1st and 3rd quartiles, with whiskers extending to the largest value less than 1.5 times the interquartile range.

We selected two experimental single-cell RNA-Seq datasets (human mature hepatocytes [8] and mouse dendritic cells [9]). On each of these datasets, we ran the six algorithms we selected for experimental datasets ten times and computed the Spearman's correlations between all 45 pairs of ranked edge lists output by these runs. We observed that every algorithm produced almost identical outputs across these multiple runs with the exception of GENIE3, GRNBoost2, and SCODE, which had a median Spearman's correlation of 0.92, 0.75, and 0.55 respectively (Supplementary Note Figure 18(a)). We also performed this analysis with all twelve algorithms for one 2,000 cell dataset for every Boolean model. While GENIE3 and GRNBoost2 had less variation for experimental single-cell RNA-Seq datasets than for simulated models, SCODE had highly varying outputs for all types of datasets (Supplementary Note Figure 18(b)).

3.5 Ensembles of GRNs

Inspired by the success of ensembles in inferring GRNs from bulk transcriptional data [10], we attempted to combine the results of the algorithms. We tried four approaches:

Borda: re-ranks every edge based on the averages of its ranks computed by each GRN inference algorithm;

mBorda: a modification of the Borda method that computes a weighted rank for each edge after down-weighting worse ranks by assigning a value of $1/n^2$ for the n^{th} ranked edge [?];

sBorda: a version of Borda that combines only the three methods with the highest AUPRC or EPR [10]; and

smBorda: a modification of mBorda that combines only the three methods with the highest AUPRC or EPR.

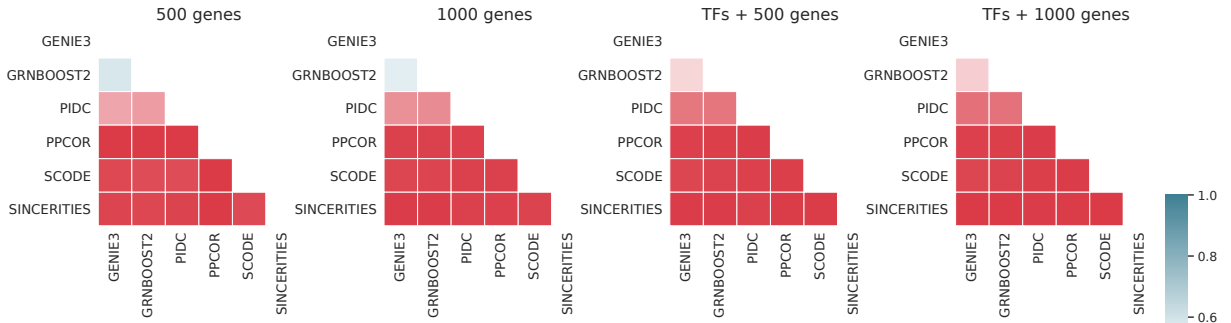
For sBorda and smBorda, we combined PIDC, GENIE2, and GRNBoost2. We performed this analysis only for the non-specific ChIP-Seq network since the STRING network contains functional (indirect) interactions.

	TFs + 500 genes							TFs + 1000 genes						
	hESC	hHep	mDC	mESC	mHSC-E	mHSC-GM	mHSC-L	hESC	hHep	mDC	mESC	mHSC-E	mHSC-GM	mHSC-L
Borda	-0.8	-0.6	-0.2	-0.7	-2.2	-1.7	0.1	-0.9	-0.7	-0.9	-0.9	-0.1	-1.6	-0.4
mBorda	-0.3	-0.4	-0.5	-0.5	-1.6	-1.5	0.2	0.3	-0.2	-0.6	-0.4	-0.4	-0.9	-0.1
sBorda	-0.2	-0.4	0.1	-0.1	-0.2	0.0	0.2	-0.3	-0.3	0.0	0.0	-0.1	0.4	-0.1
smBorda	0.2	-0.1	0.1	1.0	-0.5	-0.6	-0.4	0.1	-0.1	0.6	1.4	-0.6	-1.1	-0.4

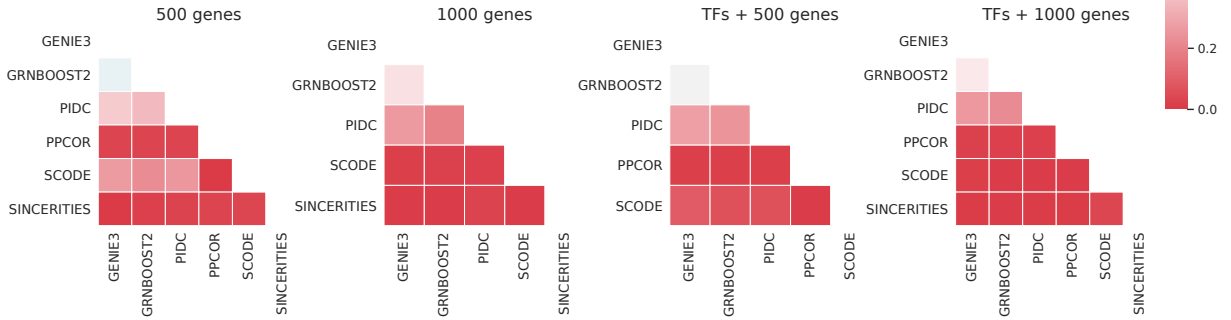
Supplementary Note Figure 19: Difference in EPR value obtained by Borda aggregation and the best performing algorithm’s EPR on the non-specific ChIP-Seq network. Each row corresponds to one of the four Borda-type methods. Each column corresponds to a experimental single-cell RNA-Seq dataset. The value in each cell is the difference in EPR value. The color of a cell is a shade of red (respectively, blue) if the difference is negative (respectively, positive).

Supplementary Note Figure 19 displays the performance of these ensembles for the experimental single-cell RNA-Seq datasets. The smBorda method was substantially better than the best performing individual algorithm for the mESC dataset [13] (both “TFs+1000 genes” and “TFs+500 genes” sets) and the mDC dataset [9] (only the “TFs+1000 genes” set) for both the non-specific ChIP-Seq and STRING networks. However, the smBorda method performed worse than or almost the same as the best individual algorithm (difference ≤ 0.1) for ten of the 14 combinations. The other three strategies (Borda, mBorda, and sBorda) either did not perform systematically better than the algorithm with the highest EPR or showed only marginal improvements. We speculate that the low similarities between the outputs of the best methods (Supplementary Note Figure 20) prevented the creation of effective ensembles.

(a) Mouse embryonic stem cells (mESC)



(b) Human definitive endoderm single-cell RNA-seq dataset (hESC)



Supplementary Note Figure 20: Heatmap of the Jaccard index for the experimental single-cell RNA-Seq dataset in (a) mouse embryonic stem cells (mESC) [11], and (b) human definitive endoderm single-cell RNA-Seq dataset (hESC) [12]. Each cell in the heatmap represents the Jaccard index between the sets of top- k edges for a pair of algorithms. We set k to be the number of edges in the intersection of the corresponding set of genes and the mouse non-specific ChIP-Seq network.

4 Software Details

In this section, we provide more information on how we executed each algorithm.

GENIE3 and GRNBoost2. We used the Python package Arboreto [14] for running GENIE3 and GRNBoost2. We choose this implementation of GENIE3 since it is the version used in an application of this method on single-cell data [15]. The current implementation of GENIE3 and GRNBoost2 in Arboreto takes an expression matrix and a list of genes as input. It produces a ranked list of edges as output, which we used directly in our evaluations.

PPCOR. We used the R implementation of PPCOR available on CRAN. The software has options to use non-parametric partial and semi-partial correlation coefficients based on Kendall’s and Spearman’s rank correlations. We used Spearman’s correlation. We used a cut-off of 0.01 on the p -values computed by PPCOR to decide whether to include an interaction in the output or not, similar to the approach used in Pseudotime Network Inference [16]. We ordered the interactions in decreasing order of absolute value of the correlation.

LEAP. We used the R implementation of LEAP. The software has options for cutoffs for maximum lag and maximum absolute correlation (MAC) and whether or not to obtain a directed network as output. We choose the option to obtain a directed network. Since we were interested in obtaining a rank for every edge, we set the MAC cutoff to 0. We searched for the best value of maximum lag as described in the “Parameter Search” section. LEAP outputs an adjacency matrix, which we converted to a ranked list of edges based on the MAC score.

SCODE. SCODE is available both as an R and a Julia package. We used the R implementation. The software has options for the number of reduced dimensions, number of iterations for optimizing ODE parameters, and for running SCODE several times. We describe how we selected these values in the “Parameter Search” section. SCODE outputs an adjacency matrix A , with positive and negative values for edge weights indicating activation or inhibition. In order to obtain the ranked list of edges, we sorted the edges by the absolute values of the edge weights.

PIDC. PIDC is available in a Julia package titled “NetworkInference”. This package has options to select various network inference algorithms such as Mutual Information, CLR, PUC and PIDC [17]. We used the maximum likelihood estimator, as recommended by the authors. We did not use any edge-weight cut-off. We converted the resulting undirected network in the form of an adjacency matrix into a ranked edge list.

SINCERITIES. SINCERITIES is available both as an R and a MATLAB package. We used the R implementation. The software has options for the distance parameter, regularization strategy, and whether or not to obtain self loops. We used the Kolmogorov-Smirnov distance and Ridge regularization as described in the SINCERITIES paper. SINCERITIES also recommends binning pseudotime values into a pre-specified number of bins. Therefore, we performed a parameter estimation on the number of bins. SINCERITIES outputs a ranked list of edges along with edge weights, which we used directly in our evaluations.

SCNS. We used the F# implementation of SCNS. SCNS requires the following inputs: binarized gene expression values for each cell, a list of cells in the initial state, a list of cells in final states, maximum numbers of activators and inhibitors for each gene, and a threshold parameter. We first binarized the gene expression data by setting the gene’s expression to zero if it was lower than average expression of that gene across all cells. We sorted the cells by pseudotime and specified that the cells in the bottom 10th percentile were in the initial state and those in the top 10th percentile were in the final state. For each gene we specified the same number of activators and inhibitors as in the reference network. We set the threshold parameter to 95. We used this value since lowering it caused SCNS to find a very large number of rules for some genes in some datasets. Moreover, the running time of SCNS made parameter search prohibitively expensive. For each gene, this method outputs a Boolean formula that combines the states (on/off) of the regulators to produce the state of the regulated gene. The formula has a specific structure: the gene is ‘on’ if and only if at least one activator is ‘on’ and all inhibitors are ‘off’. We converted each such formula into a directed network as follows: we created an edge directed from each regulator to the target gene. We ignored whether a regulator acts as an activator or inhibitor, and simply created an edge if the regulator appeared in the rule. For each gene, SCNS also outputs multiple Boolean formulae that satisfy the given input parameters. Therefore, we counted the number of rules in which a regulator appeared and used this quantity as the edge weight. We sorted this list of edges based on these weights.

SCRIBE. SCRIBE is available as an R package. Among the many variants of Restricted Directed Information presented in this paper, we used the ucRDI with $k = 1$, since this alternative appeared to have the highest accuracy in the SCRIBE manuscript; $k = 1$ means computing the RDI value conditioned on each other gene individually in order remove to indirect causal interactions. The paper says that after applying CLR, it is possible to further sparsify the network using a method for regularization. However, we did not include this step since the documentation for their R package did not give any guidelines on what functions to invoke here. SCRIBE outputs an adjacency matrix of RDI values, which we converted to an edge list.

GRISLI. GRISLI is available as a MATLAB package. GRISLI has options for parameters such as R , L and α , for which we performed parameter estimation using values recommended by the authors. GRISLI outputs a list of ranks for each edge as an adjacency matrix, which we then converted to a ranked edge list.

SINGE. SINGE is available as a MATLAB package. The software has options for various parameters, for which we performed parameter estimation. SINGE outputs a ranked list of edges along with edge weights, which we used directly in our analysis.

References

- [1] K. Street, D. Risso, R. B. Fletcher, D. Das, J. Ngai, N. Yosef, E. Purdom, and S. Dudoit. Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genomics*, 19(1):477, Jun 2018.
- [2] Clare E. Giacomantonio and Geoffrey J. Goodhill. A boolean model of the gene regulatory network underlying mammalian cortical area development. *PLoS Computational Biology*, 6(9):e1000936, 2010.
- [3] Anna Lovrics, Yu Gao, Bianka Juhász, István Bock, Helen M. Byrne, András Dinnyés, and Krisztián A. Kovács. Boolean modelling reveals new regulatory connections between transcription factors orchestrating the development of the ventral spinal cord. *PLoS ONE*, 9(11):e111430, 2014.
- [4] Jan Krumsiek, Carsten Marr, Timm Schroeder, and Fabian J. Theis. Hierarchical differentiation of myeloid progenitors is encoded in the transcription factor network. *PLoS ONE*, 6(8):e22649, 2011.
- [5] Osiris Ríos, Sara Frias, Alfredo Rodríguez, Susana Kofman, Horacio Merchant, Leda Torres, and Luis Mendoza. A boolean network model of human gonadal sex determination. *Theoretical Biology and Medical Modelling*, 12(1):26, 2015.
- [6] Vincent D. Blondel, Jean Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008.
- [7] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.
- [8] J. G. Camp, K. Sekine, T. Gerber, H. Loeffler-Wirth, H. Binder, M. Gac, S. Kanton, J. Kageyama, G. Damm, D. Seehofer, L. Belicova, M. Bickle, R. Barsacchi, R. Okuda, E. Yoshizawa, M. Kimura, H. Ayabe, H. Taniguchi, T. Takebe, and B. Treutlein. Multilineage communication regulates human liver bud development from pluripotency. *Nature*, 546(7659):533–538, 06 2017.
- [9] A. K. Shalek, R. Satija, J. Shuga, J. J. Trombetta, D. Gennert, D. Lu, P. Chen, R. S. Gertner, J. T. Gaublomme, N. Yosef, S. Schwartz, B. Fowler, S. Weaver, J. Wang, X. Wang, R. Ding, R. Raychowdhury, N. Friedman, N. Hacohen, H. Park, A. P. May, and A. Regev. Single-cell RNA-seq reveals dynamic paracrine control of cellular variation. *Nature*, 510(7505):363–369, Jun 2014.
- [10] D. Marbach, J. C. Costello, R. Kuffner, N. M. Vega, R. J. Prill, D. M. Camacho, K. R. Allison, M. Kellis, J. J. Collins, G. Stolovitzky, A. Aderhold, K. R. Allison, R. Bonneau, D. M. Camacho, Y. Chen, J. J. Collins, F. Cordero, J. C. Costello, M. Crane, F. Dondelinger, M. Drton, R. Esposito, R. Foygel, A. de la Fuente, J. Gertheiss, P. Geurts, A. Greenfield, M. Grzegorzczak, A. C. Haury, B. Holmes, T. Hothorn, D. Husmeier, V. A. Huynh-Thu, A. Irrthum, M. Kellis, G. Karlebach, R. Kuffner, S. Lebre, V. De Leo, A. Madar, S. Mani, D. Marbach, F. Mordelet, H. Ostrer, Z. Ouyang, R. Pandya, T. Petri, A. Pinna, C. S. Poultney, R. J. Prill, S. Rezny, H. J. Ruskin, Y. Saeys, R. Shamir, A. Sirbu, M. Song, N. Soranzo, A. Statnikov, G. Stolovitzky, N. Vega, P. Vera-Licona, J. P. Vert, A. Visconti, H. Wang, L. Wehenkel, L. Windhager, Y. Zhang, and R. Zimmer. Wisdom of crowds for robust gene network inference. *Nature Methods*, 9(8):796–804, Jul 2012.

- [11] T. Hayashi, H. Ozaki, Y. Sasagawa, M. Umeda, H. Danno, and I. Nikaido. Single-cell full-length total RNA sequencing uncovers dynamics of recursive splicing and enhancer RNAs. *Nat Commun*, 9(1):619, 02 2018.
- [12] Li-Fang Chu, Ning Leng, Jue Zhang, Zhonggang Hou, Daniel Mamott, David T. Vereide, Jee Choi, Christina Kendzierski, Ron Stewart, and James A. Thomson. Single-cell RNA-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm. *Genome Biology*, 17(1):173, Dec 2016.
- [13] Tetsutaro Hayashi, Haruka Ozaki, Yohei Sasagawa, Mana Umeda, Hiroki Danno, and Itoshi Nikaido. Single-cell full-length total RNA sequencing uncovers dynamics of recursive splicing and enhancer RNAs. *Nature Communications*, 9(1):619, Feb 2018.
- [14] Thomas Moerman, Sara Aibar Santos, Carmen Bravo González-Blas, Jaak Simm, Yves Moreau, Jan Aerts, and Stein Aerts. GRNBoost2 and Arboreto: efficient and scalable inference of gene regulatory networks. *Bioinformatics*, 11 2018.
- [15] S. Aibar, C. B. Gonzalez-Blas, T. Moerman, V. A. Huynh-Thu, H. Imrichova, G. Hulselmans, F. Rambow, J. C. Marine, P. Geurts, J. Aerts, J. van den Oord, Z. K. Atak, J. Wouters, and S. Aerts. SCENIC: single-cell regulatory network inference and clustering. *Nature Methods*, 14(11):1083–1086, 2017.
- [16] F. K. Hamey, S. Nestorowa, S. J. Kinston, D. G. Kent, N. K. Wilson, and B. Gottgens. Reconstructing blood stem cell regulatory network models from single-cell molecular profiles. *Proc. Natl. Acad. Sci. U.S.A.*, 114(23):5822–5829, 06 2017.
- [17] Thalia E Chan, Michael PH Stumpf, and Ann C Babbie. Gene regulatory network inference from single-cell data using multivariate information measures. *Cell systems*, 5(3):251–267, 2017.