

# Gene regulatory network structure informs the distribution of perturbation effects

Matthew Aguirre<sup>1</sup>, Jeffrey P. Spence<sup>2</sup>, Guy Sella<sup>3,4</sup>, Jonathan K. Pritchard<sup>2,5</sup>

<sup>1</sup> Department of Biomedical Data Science, Stanford University, Stanford CA

<sup>2</sup> Department of Genetics, Stanford University, Stanford CA

<sup>3</sup> Department of Biological Sciences, Columbia University, New York NY

<sup>4</sup> Program for Mathematical Genomics, Columbia University, New York NY

<sup>5</sup> Department of Biology, Stanford University, Stanford CA

Correspondence to:

magu@stanford.edu, jspence@stanford.edu,  
gs2747@columbia.edu, pritch@stanford.edu

October 22, 2024

## Abstract

Gene regulatory networks (GRNs) govern many core developmental and biological processes underlying human complex traits. Even with broad-scale efforts to characterize the effects of molecular perturbations and interpret gene coexpression, it remains challenging to infer the architecture of gene regulation in a precise and efficient manner. Key properties of GRNs, like hierarchical structure, modular organization, and sparsity, provide both challenges and opportunities for this objective. Here, we seek to better understand properties of GRNs using a new approach to simulate their structure and model their function. We produce realistic network structures with a novel generating algorithm based on insights from small-world network theory, and we model gene expression regulation using stochastic differential equations formulated to accommodate modeling molecular perturbations. With these tools, we systematically describe the effects of gene knockouts within and across GRNs, finding a subset of networks that recapitulate features of a recent genome-scale perturbation study. With deeper analysis of these exemplar networks, we consider future avenues to map the architecture of gene expression regulation using data from cells in perturbed and unperturbed states, finding that while perturbation data are critical to discover specific regulatory interactions, data from unperturbed cells may be sufficient to reveal regulatory programs.

## 22 1 Introduction

23 In the past decade, single cell sequencing assays have been instrumental in enabling functional  
24 studies of gene regulatory networks (GRNs). Observational studies of single cells have revealed  
25 substantial diversity and heterogeneity in the cell types that comprise healthy and diseased tis-  
26 sues [1], and molecular models of transcriptional systems have been used to understand the de-  
27 velopmental processes involved in maintaining cell state and cell cycle [2, 3]. Meanwhile, recent  
28 advances in the design of interventional studies, including CRISPR-based molecular perturbation  
29 approaches like Perturb-seq [4,5], have been useful for learning the local structure of GRNs around  
30 a focal gene or pathway [6, 7], discovering trait-relevant gene sets at scale [8], and determining  
31 novel functions for poorly characterized genes in a particular cell type [9]. The preponderance of  
32 single-cell data in multiple cell types, tissues, and contexts has also fueled a resurgence of interest  
33 in the wholesale inference of GRNs, capitalizing on new techniques from graph theory and causal  
34 inference [10,11].

35 In functional genomics, network inference and candidate gene prioritization are typical aims of  
36 experimental data analysis. In this setting, it is common to make assumptions about the structure  
37 and function of GRNs to enable convenient computation. In particular, linear models of gene  
38 expression on directed acyclic graphs (DAGs) have been foundational for studies of GRNs, and  
39 this approach to structure learning is well-described in the literature [12, 13]. Many extensions  
40 based on this framework have been proposed, including additional sparsity constraints in the  
41 form of regression penalties or low-rank assumptions [14, 15]. Analogous techniques have also  
42 been used in the algorithmic design of perturbation experiments [16].

43 Even though convenience assumptions like linearity and acyclicity are rarely seen as limiting  
44 in practice, it is important to note that they are not always biologically realistic. Gene regula-  
45 tion is known to contain extensive feedback mechanisms [6], and some regulatory structures (in  
46 particular, triangles, like the feed-forward motif [17, 18]) are not captured well by low-rank rep-  
47 resentations of GRNs [19]. Furthermore, biological networks are thought to be well described  
48 by directed graphs with hierarchical organization and with a degree-distribution that follows an  
49 approximate power-law [20–22]. In network inference, it is less common to make explicit use of  
50 these properties, though there are notable exceptions [7, 23].

51 With these practical considerations in mind, it is worth critically examining assumptions which  
52 are (or could be) made about the structure of GRNs. In network theory, there are well-established  
53 models of networks with group structure [24,25] and with scale-free topologies [26–28]. The defin-  
54 ing feature of directed scale-free graphs is a power-law distribution of node in- and out-degrees:  
55 this yields emergent properties including group-like structure and enrichment for structural mo-  
56 tifs [18]. Further, most nodes in these graphs are connected to one another by short paths, which  
57 is referred to as the “small-world” property of networks [29,30].

58 Here, we characterize in detail a set of structural properties that we consider to be highly rel-  
59 evant for the study of GRNs. We propose a new algorithm to generate synthetic networks with  
60 these properties and formulate a gene expression model to simulate data from them. We use this  
61 simulation framework to conduct an array of *in silico* functional genomic studies and characterize  
62 the parameter space of our model in light of a recent genome-wide Perturb-seq study [9]. Our  
63 results provide intuition about the impact of various graph properties on the susceptibility of net-  
64 works to perturbations, and on the utility of various experimental data types for inference. We

65 conclude by discussing implications for future efforts to map the architecture of gene regulation  
66 and complex traits, with particular emphasis on identifying pairwise regulatory relationships be-  
67 tween genes and clustering genes into programs. Our analysis tools are available on github as a  
68 resource to the scientific community.

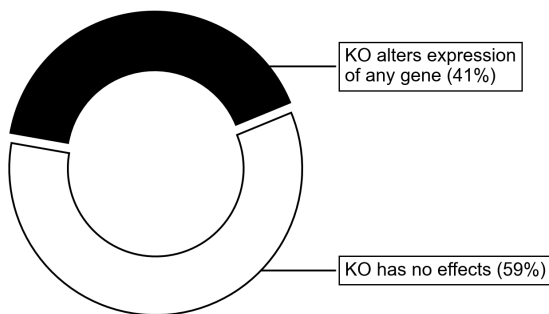
## 69 2 Main

### 70 2.1 Modeling approach

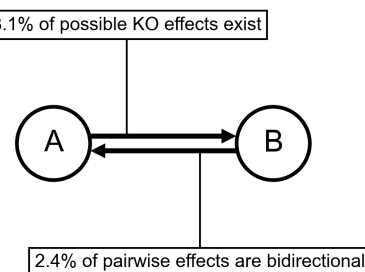
71 Inspired by previous work from network theory and systems biology, we list what we consider  
72 to be key properties of GRNs. We motivate these criteria in light of a recent genome-scale study  
73 of genetic perturbations, conducted in an erythroid progenitor cell line (K562) (**Fig. 1**) [9]. To  
74 date, this is one of the largest available single-cell and single-gene perturbation datasets in any  
75 cell type: the data contain measurements on the expression of 5,530 gene transcripts in 1,989,578  
76 cells, which were subject to 11,258 CRISPR-based perturbations of 9,866 unique genes. Here, we  
77 subset these data to 5,247 perturbations that target genes whose expression is also measured in  
78 the data (**Methods**). Key network properties are as follow:

- 79 1. **GRNs are sparse:** While gene expression is controlled by many variables, the typical gene is  
80 directly affected by a small number of regulators. We further expect the number of regulators  
81 of any single gene to be much smaller than the total number of regulators in the network.  
82 Also, not all genes participate in expression regulation: only 41% of perturbations that target  
83 a primary transcript have significant effects on the expression of any other gene (**Fig. 1A**).
- 84 2. **GRNs have directed edges and feedback loops:** Regulatory relationships between genes  
85 are directed, with one gene acting as a regulator and the other as a target gene: this means  
86 that “A regulates B” is distinct from “B regulates A”. Meanwhile, feedback loops are also  
87 thought to be pervasive in gene regulatory networks. A simple case of a feedback loop is  
88 bidirectional regulation, which is observed in data: 3.1% of ordered gene pairs have at least  
89 a one-directional perturbation effect (i.e., “A affects B”, Anderson-Darling FDR-corrected  
90  $p < 0.05$ ), and 2.4% of these pairs further have bi-directional effects (i.e., “B also affects A”)  
91 (**Fig. 1B**).
- 92 3. **GRNs have asymmetric distributions of in- and out-degree:** A further asymmetry between  
93 regulators and target genes arises from the existence of master regulators, which directly par-  
94 ticipate in the regulation of many other genes. The number of regulators per gene and genes  
95 per regulator are both thought to follow an approximate power-law distribution [20, 21],  
96 and indeed, the number of perturbation effects per regulator has a heavier-tailed distribu-  
97 tion than the number of effects per target gene (**Fig. 1C**).
- 98 4. **GRNs are modular:** Genes in regulatory networks have different molecular functions that  
99 are executed in concert across various cell and tissue types. This grouping of genes by func-  
100 tion also corresponds to a hierarchical organization of regulatory relationships that is re-  
101 vealed when these programs respond similarly to certain sets of perturbations (**Fig. 1D**).

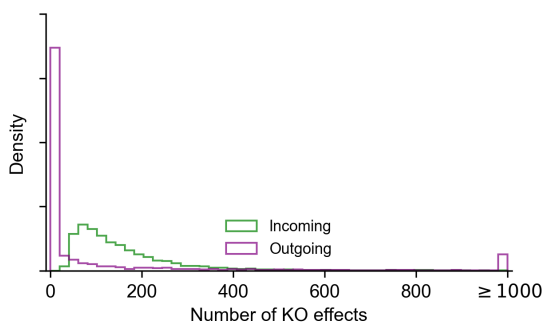
A: Perturbation effects are sparse



B: Edges are directed and can form loops



C: Degree distributions are asymmetric



D: Genes are organized into modules

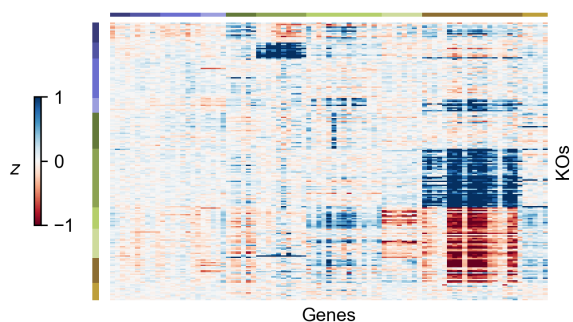


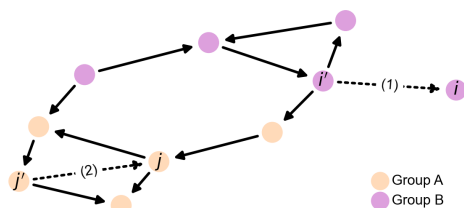
Figure 1: **Key properties of gene regulatory networks.** Data from Replogle *et. al.*, 2022. **(A)** Of the 5,247 perturbations in our analysis subset, 2,152 (41%) have a measurable effect on the transcriptional state of cells (energy-test  $p < 0.001$ ). **(B)** Among all ordered pairs of genes, 3.1% (865,719 pairs) have a one-directional effect (FDR-corrected  $p < 0.05$ ). Of these pairs with KO effects, 2.4% (20,621 pairs) further have bidirectional effects. **(C)** Summaries of the distribution of KO effects (Anderson-Darling  $p < 0.05$ ) from the perspective of genes as subject to perturbation (outgoing effects) and as target genes when other genes are perturbed (incoming effects). **(D)** Subset of  $z$ -normalized expression data corresponding to 10 gene modules, using labels as provided in the dataset – each modules is labeled by a color in the to bars above the  $x$ - and  $y$ -axes, and  $z$ -scores are clipped at  $\pm 1$ , for visualization.

102 While these criteria are not exhaustive, they do substantially constrain the space of plausible  
103 GRN structures. But from first principles, it is not obvious why GRNs may have these properties,  
104 or whether these properties might also limit what can be discovered from experimental data. In  
105 other words, what does it matter that networks are sparse, or modular?

## 106 2.2 Network generating algorithm

107 To better understand these foundational questions about the impacts of gene regulatory network  
108 architecture, we propose a two-step process to simulate synthetic GRNs. First, we produce re-  
109 alistic graph structures using a novel generating algorithm: we show that its parameters control  
110 key properties of the resulting graphs. Second, we describe a dynamical systems model of gene  
111 expression, which we use to generate synthetic data from arbitrary graph structures. With these  
112 tools, we conduct an array of simulated molecular perturbation studies, varying network proper-  
113 ties of interest: an overview of our network generating algorithm is in **Fig. 2A**.

### A: Overview of network generating algorithm



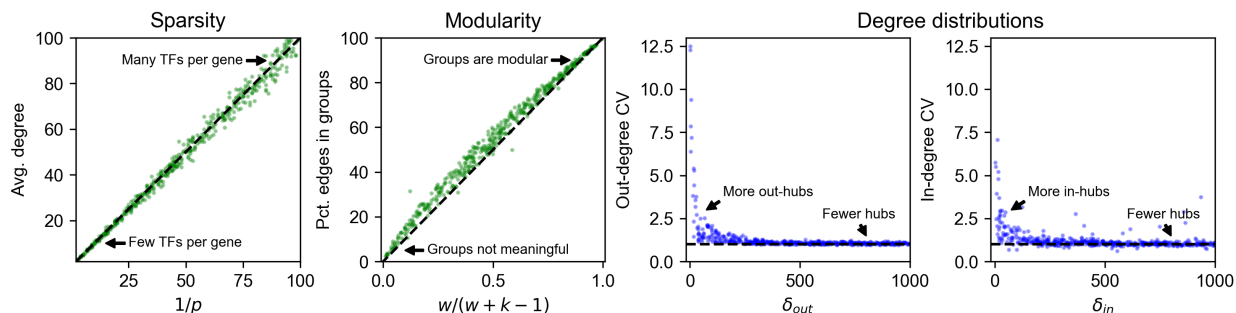
Parameters for 1,920 GRNs in this study:

- Number of genes ( $n$ ): 2000
- Number of groups ( $k$ ): 1, ..., 100
- Sparsity term ( $p$ ): 1/2, ..., 1/16
- In-group term ( $w$ ): 1, ..., 900
- In-degree term ( $\delta_{in}$ ): 10, ..., 300
- Out-degree term ( $\delta_{out}$ ): 1, ..., 30

Perform one of these two steps at random, until the graph has  $n$  nodes:

- (1). With prob.  $p$ : Attach a new node ( $i$ ) with an incoming edge. Pick the source ( $i'$ ) using out-degrees ( $\delta_{out}$ ) and groups ( $w$ ).
- (2). With prob.  $1 - p$ : Add a new edge. Pick the target ( $j$ ) using in-degrees ( $\delta_{in}$ ), and the source ( $j'$ ) using out-degrees ( $\delta_{out}$ ) and groups ( $w$ ).

### B: Graph properties controlled by algorithm parameters



**Figure 2: Modeling approach and network generating algorithm.** (A) Overview of network generating algorithm, based on a growth process with preferential attachment. At each step, randomly add either a node or an edge, with the source and target determined by the out- and in-degree distributions, and node membership in groups. (B) Key graph properties can be tuned by changing the parameters of the generating algorithm. We validate this in 1,000 synthetic graphs with 500 nodes each, produced with various generating parameters. The same networks are plotted in all four panels, indicating robustness across different background distributions of parameters.

114 Our algorithm is based on that of Bollobás *et. al.*, 2003 [28], which models network growth  
 115 with preferential attachment. This algorithm starts with a small initial graph, and randomly adds  
 116 nodes or directed edges until the graph reaches a pre-specified size. When adding a node, the new  
 117 node is selected to be the target of a new directed edge. When adding an edge between existing  
 118 nodes, a node is selected to be the target with a probability that increases with the number of  
 119 outgoing edges it already has. When selecting a node to be the source of a new edge (i.e., to be  
 120 the regulator for a new gene, if we are adding a node, or for an existing gene, if we are adding an  
 121 edge), we select with probability increasing in the number of incoming edges it already has. Our  
 122 work generalizes this algorithm in two ways: first, by assigning each node in the network to one of  
 123 a number of pre-specified groups, and second, by specifying a within-group affinity term which  
 124 biases edges to be drawn between members of the same group. The full procedure, including  
 125 pseudocode and a description of its parameters, is given by **Algorithm 1**.

126 The output of our algorithm is a directed scale-free network on  $n$  nodes, each of which is  
 127 assigned to one of  $k$  groups. The parameters in our algorithm control specific network properties.  
 128 To show this, we generated 1,000 synthetic graphs with  $n = 500$  genes using an array of randomly  
 129 sampled parameters (**Fig. 2B**). We observe that the sparsity term,  $p$ , adjusts the mean number

---

### Algorithm 1 Directed scale-free network with groups

---

**Require:**

- $n$ : Number of genes (nodes) in the network ( $n \geq 3$ ).
- $k$ : Number of groups in the network ( $1 \leq k \leq n$ ).
- $p$ : Sparsity term ( $0 < p \leq 1$ ).
- $\delta_{in}$ : In-degree biasing term ( $\delta_{in} \geq 0$ ).
- $\delta_{out}$ : Out-degree biasing term ( $\delta_{out} \geq 0$ ).
- $w$ : Group biasing term ( $w \geq 1$ ).

```

▷ Initialize the graph  $G$  to be a three-node cycle. Assign each node to its own group. ◁
 $G \leftarrow \{(1 \rightarrow 2), (2 \rightarrow 3), (3 \rightarrow 1)\}$ 
 $gp(i) \leftarrow i, \quad i \in \{1, 2, 3\}$  ◁ If  $k = 2$  then assign node 3 to group 1.

▷ Grow the graph  $G$  according to the below steps, until it has  $n$  genes. ◁
while  $|G| < n$  do
  ▷ Pick a gene (node)  $i$  to be the target of a new regulatory relationship (edge). With probability  $p$ ,
  add a new gene to  $G$ , otherwise pick an existing gene proportional to the in-degree distribution. ◁
  if  $\text{runif}(0, 1) < p$  then
     $i \leftarrow |G| + 1$ 
     $gp(i) \leftarrow g \in \{1, \dots, k\}$  uniformly at random.
  else
     $i \leftarrow i \in \{1, \dots, |G|\}$  with probability  $p_i \propto \text{in-degree}(i) + \delta_{in}$ 
  ▷ Then pick a gene (node)  $j$  to regulate  $i$ , proportional to the out-degree distribution, weighted by
  whether  $i$  is in the same group as  $j$ . ◁
   $j \leftarrow j \in \{1, \dots, |G|\}$  with probability  $p_j \propto (\text{out-degree}(j) + \delta_{out}) \times (w \text{ if } gp(i) = gp(j) \text{ else } 1)$ 
  ▷ Add the edge ( $j \rightarrow i$ ) to the graph. Note that  $j$  and  $i$  may be the same node, in which case we
  add the edge ( $j \rightarrow j$ );  $j$  and  $i$  may also already share the edge ( $j \rightarrow i$ ), in which case we add a
  duplicate edge. ◁
   $G \leftarrow (j \rightarrow i)$ 

```

---

130 of regulators per gene, which is approximately  $1/p$  (**Fig. 2B**). The number of groups,  $k$ , and the  
 131 modularity term,  $w$ , determine the fraction of edges which are drawn between members of the  $k$   
 132 groups – this fraction is approximately  $w/(w + k - 1)$  (**Fig. 2B**). Finally, the bias terms  $\delta_{in}$  and  $\delta_{out}$   
 133 respectively control the coefficient of variation (CV) of the in- and out-degree distributions (**Fig.**  
 134 **2B**). CV is the standard deviation of a distribution over its mean, and for power-law distributions  
 135 the CV is related to the power-law coefficient: a larger CV means the distribution has a heavier tail  
 136 (i.e. there are hub regulators which have many target genes; or there are hub target genes which  
 137 are directly affected by many regulators).

### 138 2.3 Expression model

139 In order to enable reasonable comparisons with experimental data, we use an expression model  
 140 with quantitative (rather than binary) measurements, and with dynamics subject to a non-linearity

141 that enforces realistic physical constraints: gene expression is non-negative and saturates near a  
142 maximum value. Given a graph structure generated using the algorithm above, we assign pa-  
143 rameters to each gene (node) and regulatory interaction (edge) in the graph. Each gene  $i$  has two  
144 rate parameters: one for innate RNA production in the absence of regulators ( $\alpha_i$ ), and another for  
145 the decay of existing cellular RNAs ( $\ell_i$ ). Each regulatory relationship, between genes  $j$  and  $i$ , has  
146 one parameter: a magnitude ( $\beta_{ji}$ ) which describes the importance of the regulator for the expres-  
147 sion of the target gene. We also enforce a constraint that interactions have a minimum strength  
148 ( $|\beta_{ji}| \geq 1$ ). A full description of the strategy we use to sample these parameters for synthetic  
149 GRNs is in **Methods**.

150 Our expression model takes the form of a stochastic differential equation (SDE), and we pro-  
151 duce expression values using forward simulation according to the Euler-Maruyama method (**Fig.**  
152 **3A**). For gene  $i$  with regulators  $j$  having expression  $x_j$  (likewise  $x_i$ ) at time  $t$ , the difference equation  
153 for expression  $x'_i$  at time  $t + \Delta t$  is given by

$$\frac{x'_i - x_i}{\Delta t} = \sigma(\alpha_i + \sum_j x_j \beta_{ji}) - \ell_i x_i + \mathcal{N}(0, s^2 \frac{x_i}{\Delta t}),$$

154 where the terms on the right hand side of the equation, in order, correspond to transcriptional  
155 synthesis, degradation, and noise. Unless stated otherwise, we set  $s = 10^{-4}$  as the magnitude of  
156 noise, which serves to scale the intrinsic biological noise in synthesis and degradation of RNAs  
157 (hence noise is also proportional to  $x_i$ ). We let  $\Delta t = 0.01$  be the step size, as in previous work [31],  
158 and take  $\sigma(x)$  as the logistic sigmoid (expit) function  $\sigma(x) = 1/(1 + e^{-x})$ . In practice, we conduct  
159 forward simulation in vectorized form with an update rule:

$$x' = x + \Delta t \cdot (\sigma(\alpha + \beta^\top x) - \ell x) + \mathcal{N}(0, \Delta t \cdot s^2 \text{diag}(x)).$$

160 Throughout our experiments, we perform on the order of thousands of iterations and then check  
161 that the system of differential equations has reached an expression steady-state (**Methods**).

162 Our model can be used to quantify the effects of many types of perturbations. These include  
163 (1) gene knockouts (KOs), which we model by nullifying  $x_j = 0$  (or equivalently, setting  $\beta_{ji} = 0$  for  
164 all  $i$ ); (2) gene knockdown or overexpression, which can be modeled by decreasing or increasing  
165  $\alpha_j$ , increasing or decreasing  $\ell_j$ , or directly manipulating  $x_j$  to a fixed value; (3) enhancer edits or  
166 transcriptional rewiring, modeled by changing specific  $\beta_{ji}$ ; and (4) changes to expression noise,  
167 modeled by altering  $s$ , either globally or for specific genes. We further note that similarly for-  
168 mulated perturbations with small magnitudes could also make appropriate models of the effects  
169 of molecular quantitative trait loci (QTLs). Here, we focus solely on gene knockouts, which we  
170 consider for the remainder of this work.

## 171 2.4 Perturbation studies

172 We conducted synthetic perturbation studies in 1,920 GRNs with  $n = 2,000$  gene; these GRNs  
173 were produced with a range of network generating parameters (**Methods**). For each GRN, we  
174 initialized gene expression values at zero and conducted a minimum 5,000 iterations of forward  
175 simulation, later verifying that the dynamical system reached equilibrium and assessing its stabil-  
176 ity (**Fig. 3A,B, Methods**). We then independently knocked out each gene in the network and let

177 the system re-equilibrate after additional rounds of forward simulation (**Fig. 3B**). We computed  
178 the effect of perturbing gene  $j$  as the log-fold change in expression  $x_i$  of all other genes  $i$ , which  
179 we refer to as the “perturbation effect” of gene  $j$  on gene  $i$ . Mathematically, this is

$$\log_2 \text{FC}_{ji} = \log_2(x_i | \text{do}(x_j = 0)) - \log_2(x_i)$$

180 where  $x_i | \text{do}(x_j = 0)$  denotes the expression of gene  $i$  when gene  $j$  has been knocked out. While the  
181 majority of perturbation effects are small in all GRNs, with 86.6% of all effects below  $|\log_2 \text{FC}| =$   
182 0.01, each network harbors a median 5,296 large effects on the order  $|\log_2 \text{FC}| = 1$  (**Fig. 3C**). We  
183 also note substantial variability in the distribution of perturbation effects across networks (**Fig.**  
184 **3C**).

185 These effects are largely stratified by the distance between regulator and target (**Fig. 3D**), with  
186 distance here being the length of the shortest path between genes along edges in the network.  
187 Across GRNs, a majority of direct regulators of a gene confer at least a modest effect on average  
188 (77.3% of genes at distance 1 have  $|\log_2 \text{FC}| > 0.01$  when knocked out). Meanwhile, indirect  
189 effects of this magnitude also exist, but are less common on a per-interaction basis (mean 21.5%  
190 of gene pairs not connected by an edge). However, since mediation is much more common than  
191 direct regulation, mediated effects contribute a substantial fraction of perturbation effects at all  
192 but the largest magnitudes – for example, 98.5% of effects at  $|\log_2 \text{FC}| > 0.01$  across GRNs are  
193 mediated rather than direct (**Fig. S1**).

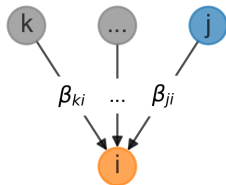
194 Since genes in the simulated GRNs belong to pre-defined groups, we further investigated the  
195 extent to which perturbation effects cluster within rather than across groups. On average, there is  
196 an enrichment of effects within groups – but as with the overall distribution, there is heterogeneity  
197 in the distributions of within- and across-group perturbation effects (**Fig. 3E**). This heterogeneity  
198 is driven largely by the modularity term: as  $w$  increases, the distributions of within- and across-  
199 group effects become further separated, even across networks with different numbers of groups  
200 (**Fig. S2**). This effect is based on changes in network architecture: since the strongest perturbation  
201 effects are from direct regulators, an increased affinity for within-group regulation (i.e., larger  $w$ )  
202 means that these effects should also come from members of the same group.

## 203 **2.5 Impact of network properties**

204 Next, we turned our attention to the relationship between properties of networks (as determined  
205 by network generating parameters) and their distributions of perturbation effects. As summaries  
206 of this distribution, and of the overall susceptibility of individual GRNs to perturbations, we com-  
207 pared the number of genes which are hub KOs and hub target genes in each of the 1,920 synthetic  
208 GRNs. We say a gene is a hub KO if it introduced a change of  $|\log_2 \text{FC}| > 0.1$  in at least 100 other  
209 genes when knocked out; analogously, we say a gene is a hub target gene if its expression was  
210 changed by  $|\log_2 \text{FC}| > 0.1$  upon knockout of at least 100 other genes. Genes whose equilibrium  
211 expression was below the magnitude of noise were removed from these counts, as their expres-  
212 sion could vary widely across conditions solely due to noise. We find that these statistics behave  
213 consistently with respect to the network generating parameters across GRNs (**Fig. 4**), and that the  
214 directions of effect are also similar to the overall number of perturbation effects at this threshold  
215 in the network (**Fig. S3**).



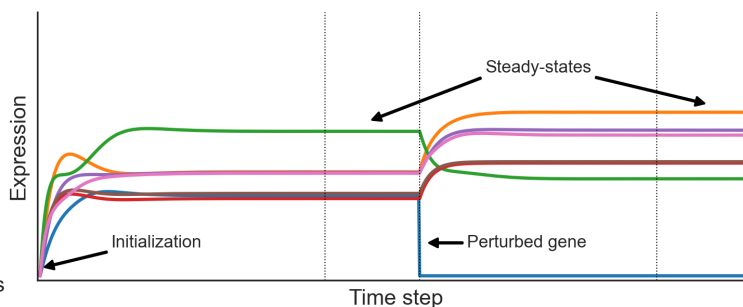
A: Model gene expression



Stochastic differential equation (SDE) has terms for synthesis, degradation, and noise for gene  $i$  at times  $t$ ,  $t+\Delta t$ :

$$\frac{x_i^{[t+\Delta t]} - x_i^{[t]}}{\Delta t} = \sigma\left(\alpha_i + \sum_k \beta_{ki} x_k^{[t]}\right) - \ell_i x_i + s \sqrt{\frac{x_i^{[t]}}{\Delta t}} \mathcal{N}(0, 1)$$

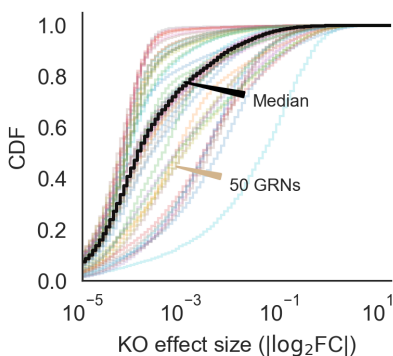
B: Compute gene knockout (KO) effects



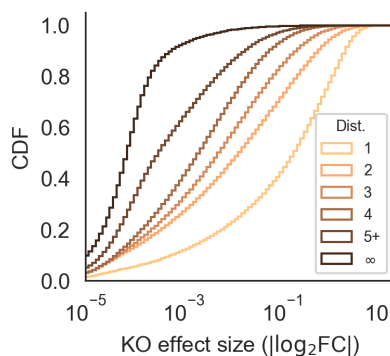
KO effect sizes are  $\log_2$  fold-changes between expression steady-states:

$$\log_2 FC_{ji} = \log_2(x_i | d_0(x_j = 0)) - \log_2(x_i)$$

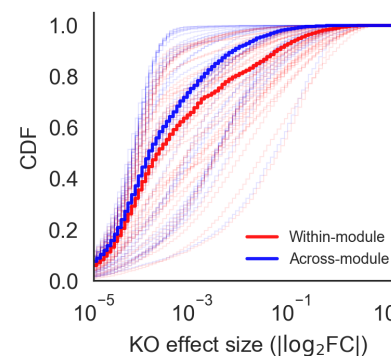
C: KO effects in 50 GRNs



D: KO effects by graph distance



E: KO effects by module



**Figure 3: Perturbations and their effects within networks.** (A) Overview of gene expression model and its parameters. Here,  $\sigma$  is the logistic sigmoid  $\sigma(x) = 1/(1 + e^{-x})$ . (B) Example forward simulation of the dynamical systems model. Trace lines show genes, whose expression values are initialized at zero. The system eventually reaches a steady-state, and is then subject to perturbation (knockout of gene  $j$ , i.e. holding  $x_j = 0$ ). Further forward simulation leads to a new steady-state, from which we can compute perturbation effects ( $\log_2$  FC for other genes  $i$ ). (C) Distribution of knockout (KO) effects (i.e.,  $\log_2$  fold-changes in expression  $x_i$  of a focal gene  $i$ ) in 50 example GRNs, along with the median distribution (black line). (D) KO effects as a function of network distance between two genes, and (E) within and across modules given by the generating algorithm. Note that the solid lines in (D) and (E) are the median distributions over the 50 example GRNs, split respectively by distances and modules.

216 Graph sparsity has the greatest influence on the number of hub KOs and hub target genes in  
 217 the GRNs (Fig. 4A). More regulators per gene (large  $1/p$ ) tends to translate to more perturbation  
 218 effects overall, increasing the number of both hub knockouts and target genes. Notably, the effects  
 219 on regulators and targets are not identical. In denser networks, the median number of hub KOs  
 220 tends to be larger than the number of hub target genes. However, in a subset of dense networks,  
 221 most genes in the network are identified as hub target genes. This is related to the absence of  
 222 stable equilibrium dynamics in the low-noise limit of the gene expression model (Fig. S4), which  
 223 suggests that as GRNs become more dense and genes are subject to regulation by larger fractions  
 224 of the network, the system is less likely to be stable. Although this term has a large effect on  
 225 perturbation effects, we find no obvious interaction between it and other terms in the generating

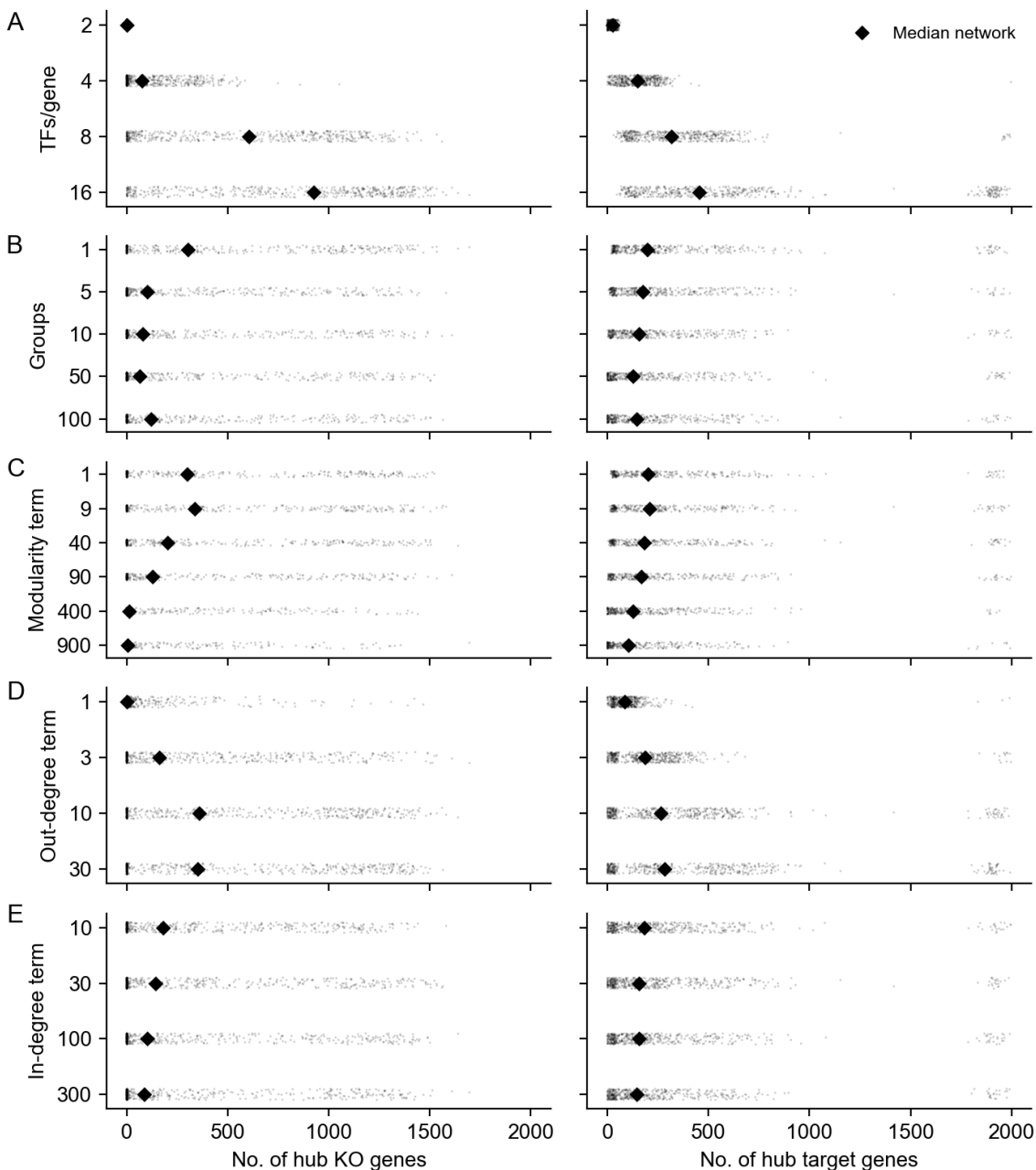


Figure 4: **Network properties influence their susceptibility to gene perturbations.** Counts of genes that are hub knockouts (**left**) and hub target genes (**right**) in each synthetic GRN, as a function of network generating parameters. Each panel shows all 1,920 GRNs as individual points, stratified by parameter values. Each distribution is annotated with its mean over GRNs (diamond points).

226 algorithm (**Fig. S5**).

227 GRNs with fewer groups (small  $k$ ) and higher modularity (large  $w$ ) tend to have fewer hub  
 228 KOs and hub target genes (**Figs. 4B, 4C**). The modularity term monotonically increases resilience  
 229 to perturbation; the group term monotonically decreases it, with the exception of  $k = 1$ . From

230 the perspective of the network generating algorithm,  $k = 1$  and  $w = 1$  are identical; they are  
231 equivalent to the algorithm from Bollobás *et. al.*, 2003 [28] and correspond to the dissolution of  
232 modular structure with respect to the specified grouping. This is also equivalent to  $k = 2000$ , in  
233 which each gene in the network has its own group – this intuition is supported by the remaining  
234 trend in (Fig. 4B). Meanwhile, in modular networks (large  $w$ ), most edges are between members  
235 of the same group (Fig. 4C). This might serve to confine the downstream effects of perturbations  
236 to members of the same group, effectively dampening the transcriptional impact of altering the  
237 function of master regulators.

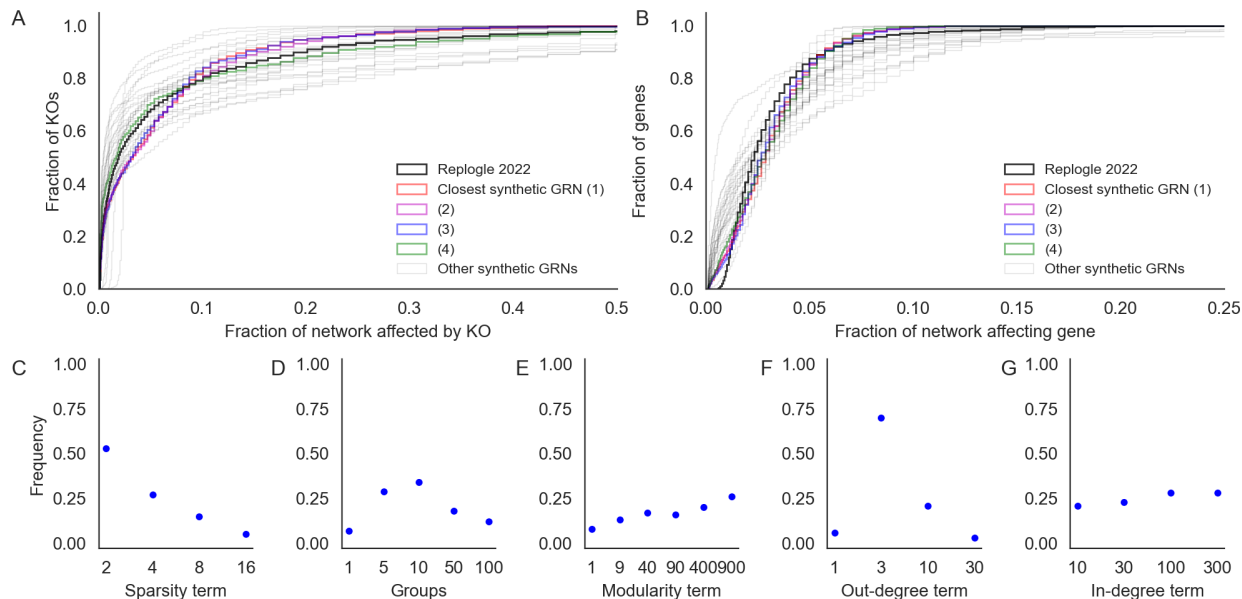
238 When the out-degree distribution of GRNs has a heavier tail (small  $\delta_{out}$ ), there tend to be many  
239 fewer hub knockout genes (Fig. 4D). This relationship is non-linear, and in the most extreme case  
240 ( $\delta_{out} = 1$ ) there are only 89 hub KOs on average (median 1 hub KO) in the GRN. This effect is  
241 a consequence of preferential attachment: as more edges are drawn from master regulators, out-  
242 going regulatory effects also concentrate there as well. Counterintuitively, this parameter exerts  
243 influence over the number of hub target genes in the network as well, and in the same direction.  
244 When effects are concentrated among a few key regulators, it may simply be less feasible for any  
245 gene to be affected by many knockouts since there are fewer genes that have many knockout ef-  
246 fects at all. As with the sparsity term, we do not see obvious interactions between this term and  
247 others in the generating algorithm (Fig. S6). Meanwhile, when the in-degree distribution of GRNs  
248 has a less heavy tail (large  $\delta_{in}$ ), there are modestly more hub target genes and hub KOs (Fig. 4E).  
249 The source of this trend is difficult to intuit, but the effect is very weak.

250 Looking across parameters, these results reflect a wide range of variation in the susceptibility  
251 of GRNs to perturbations as a function of their structural properties. While there is substantial  
252 overlap in the distributions of hub KO and target genes across network generating parameters,  
253 we find that all parameters except the in-degree term have statistically significant effects on both  
254 quantities ( $p < 0.001$  for all tests, Fig. S7 – full results in Tables S1, S2). We estimate these effects  
255 with a multiple regression on the logit-transformed fraction of genes in each GRN which are hub  
256 KOs or hub target genes (Methods). In total, the network generating parameters explain just un-  
257 der half the variance in the fraction of the GRN which is either a hub KO (model  $r^2 = 0.59$ ) or hub  
258 target gene (model  $r^2 = 0.46$ ). Moreover, there is also a noteworthy thematic consistency across  
259 parameters. The direction of protective effect from perturbation was consistently that of biological  
260 realism: GRNs with fewer hub KOs and target genes tended to be sparser, more modular, and had  
261 a heavier tailed out- but not in-degree distribution.

## 262 2.6 Comparing with experimental data

263 With this intuition about network properties in hand, we now return to real experimental data.  
264 Given that synthetic GRNs with quantifiably different structures produce qualitatively different  
265 distributions of knockout effects, we next sought to ask whether any of them were also similar  
266 to real data. For this, we made use of the subset of perturbations that correspond one-to-one  
267 with gene expression readout in a recent Perturb-seq study [9]. We use data from this study  
268 in particular for two reasons: one being its scale (thousands of genes measured), and the other  
269 its unbiased selection of genes to perturb. These features contrast with prior work focused on  
270 specific pathways [6] or on genes expected to have large transcriptional effects when perturbed  
271 (e.g., “essential” gene experiments from the same study [9]). This scale and unbiasedness is unique

272 in the literature, and we find them to be critical for understanding global features of regulatory  
 273 networks at the resolution we seek to model.



**Figure 5: Comparing with genome-wide Perturb-seq.** Fraction of GRN that (A) affects each gene when perturbed (outgoing effects) or (B) is affected by other perturbations (incoming effects). In synthetic data, perturbation effects are thresholded at the top 3% of absolute log-fold change values, matching the proportion of pairwise tests from the Perturb-seq data with FDR-corrected Anderson-Darling  $p < 0.05$ . Highlighted in color are the four GRNs that best match the Perturb-seq data. (C-G) Distribution of network generating parameters for the 100 GRNs that are best matched to Perturb-seq data (by Kolmogorov-Smirnov  $p$ -value rank for both distributions in A and B).

274 We compared the real and synthetic data using their cumulative distributions of perturbation  
 275 effects, computed both from the perspective of genes as regulators and as targets of regulation  
 276 (Fig. 5A-B). Since these data have different numbers of genes that are not lowly expressed, we  
 277 normalized the number of incoming and outgoing effects to the size of the network (Methods). In  
 278 the Perturb-seq data, we find noticeable qualitative differences between the distribution of incom-  
 279 ing and outgoing effects. These differences are consistent with the notion that GRNs should have  
 280 master regulators, but not master target genes. In the synthetic data, we find substantial diversity  
 281 in both distributions across GRNs, including many patterns that seem wholly incompatible with  
 282 those observed in experiments. Meanwhile, some GRNs seem well-matched to the Perturb-seq  
 283 results: the distributions closest to the data are highlighted in color in Fig. 5A-B, and correspond  
 284 to those having the smallest Kolmogorov-Smirnov test statistics when compared with the data  
 285 distributions (Methods).

286 Although the focus of our work is not wholesale network inference, we do observe a coherent  
 287 set of properties among the well-matched networks (Fig. 5C-G). Specifically, they share a rela-  
 288 tively small number of regulators per gene (two, rather than 16); they have a small number of  
 289 groups (five to ten rather than one or 100); they are highly modular (large  $w$ ); and they have a  
 290 heavy tail in the distribution of out-degree but not in-degree ( $\delta_{out}$  near three but  $\delta_{in}$  on the order  
 291 of 100). Consistent with previous results, we find these parameter sets to be within a range that

292 matches our motivating intuition about the structural properties of GRNs. We further do not find  
293 these properties to have obvious pairwise interactions that affect concordance to data (**Fig. S8**).

294 It is worth noting that while some of the networks seem well matched to real data by eye, all  
295 of them are statistically distinguishable from the observed distributions of incoming and outgo-  
296 ing effects (**Supplementary Data 1**). This is expected behavior since the networks we generate  
297 are entirely synthetic, leveraging no prior biological information about the structure of the human  
298 genome or statistical properties of this specific dataset. To these points, we suspect that a more  
299 flexible specification of group size, number, and affinity (i.e., altering the  $w$  parameter) could im-  
300 prove concordance with the distribution of outgoing effects (**Fig. 5A**), and that modeling the  
301 detection power of experimental assays could improve concordance with the distribution of in-  
302 coming effects (**Fig. 5B**), especially seeing as baseline expression is a stronger determinant of the  
303 number of detected incoming effects in real data compared to simulations (likely due to power  
304 limitations in the perturb-seq data; **Fig. S9**). Despite these limitations, we find that all of the pa-  
305 rameters of our algorithm are necessary to match the observed distribution from data. Notably, it  
306 is not sufficient to let group structure be an emergent property of the network (as in Bollobás *et*.  
307 *al.* 2003), which corresponds to  $k = 1$  (or  $w = 1$ ) under our model. Networks generated by the  
308 original Bollobás algorithm do not match real data as well as those with enforced group structure  
309 (**Fig. 5 D,E**).

## 310 **2.7 Challenges and opportunities for inference**

311 Finally, we highlight the utility of our simulation approach by considering the value of different  
312 data sources for inference tasks. For this, we conducted further analysis using an example syn-  
313 thetic GRN whose patterns of knockout effects were well-matched with Perturb-seq data. Specif-  
314 ically, we focused on the recovery of edges, edge weights, and group structure using interven-  
315 tional data (e.g., perturbation effects) and observational data (e.g., coexpression). We made use  
316 of perturbation effects as previously described, and further computed pairwise gene coexpression  
317 values using additional rounds of forward simulation from the expression model at steady state  
318 to approximate the naturally occurring variation across cells (**Methods**).

### 319 **2.7.1 Discovering pairwise relationships**

320 Several computational and experimental approaches have been used to estimate pairwise causal  
321 relationships between genes, with the ultimate goal of wholesale inference of gene regulatory net-  
322 works [10, 11]. These data and methods are broad in scope, and range from estimating networks  
323 using natural variation in gene expression values from bulk tissue [12, 23] to fitting complex ma-  
324 chine learning models on data from single-cell perturbation experiments [7, 15, 32]. Here, we  
325 consider two descriptive pairwise summary statistics at the gene level – gene coexpression across  
326 cells and perturbation effects across gene knockouts – and their connections to edges in a simu-  
327 lated GRN.

328 In the synthetic data, we find that the distributions of pairwise gene coexpression values and  
329 knockout effects both span multiple orders of magnitude (**Fig. 6**). However, where the distribu-  
330 tion of knockout effects differs dramatically between gene pairs that share an edge and those that  
331 do not, the distributions of coexpression values have substantial overlap (**Fig. 6A,B**). This differ-

332 ence in distribution reflects what each statistic tends to measure. Gene perturbation effects tend  
 333 to flow through the network along edges, and are therefore highly related to the network distance  
 334 between genes (**Fig. 3D**). This includes whether or not two genes share a direct regulatory rela-  
 335 tionship in the form of an edge (**Fig. S10**). Meanwhile, strong coexpression is more often due to  
 336 co-regulation than to there being a direct causal relationship between genes (**Fig. S10**), making it  
 337 difficult in practice to uncover edges with this data.

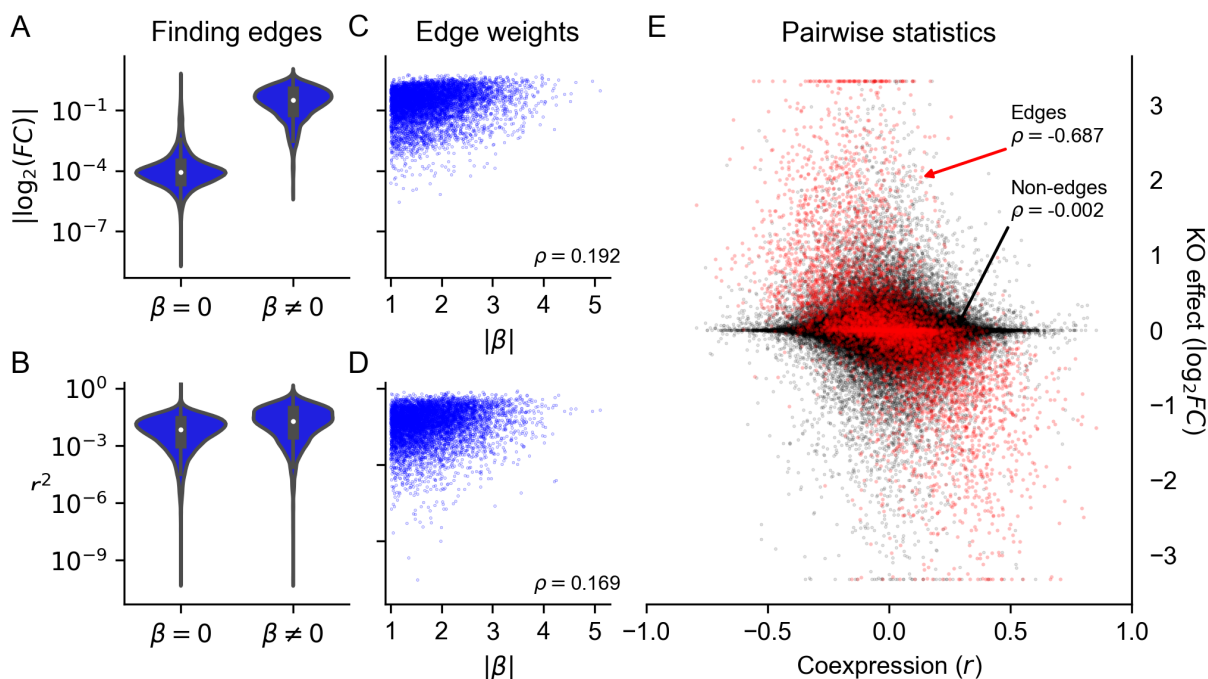


Figure 6: **Perturbations more reliably measure fine-scale network structure than coexpression.** (A) Distribution of perturbation effects and (B) coexpression values between pairs of genes that do or do not share an edge in a realistic synthetic GRN (both pairs of distributions have statistically distinguishable means;  $p < 10^{-16}$ , two-sample  $t$ -test). (C) Rank correlation of perturbation effect sizes and (D) coexpression magnitude with edge weights. (E) Rank correlation between coexpression and perturbation effects (the  $y$ -axis is clipped at values corresponding to tenfold change).

338 For gene pairs where there is direct regulation we see that both knockout effects (**Fig. 6C**) and  
 339 coexpression (**Fig. 6D**) have weak correlation with the strength of known regulatory relationships.  
 340 This reflects that both statistics are imperfect measures of regulatory importance: they are both  
 341 affected by differences in regulatory architecture across genes (e.g., number of regulators or the  
 342 intensity of transcriptional buffering). We further see this when directly comparing coexpression  
 343 and knockout effects between pairs of genes. Across all pairs of genes, these two statistics are  
 344 uncorrelated – but the two are highly correlated among pairs of genes that share an edge (**Fig.**  
 345 **6E**). In this way, both perturbation effects and coexpression contain similar information about  
 346 edges in the GRN – but coexpression also measures non-causal relationships between genes, like  
 347 coregulation, and is therefore systematically uncorrelated with perturbation effects even in real  
 348 data (**Fig. S11**).

349 We might therefore expect that network features associated with increased coregulation (like  
 350 modularity and the presence of transcriptional master regulators) will widen the gap in perfor-

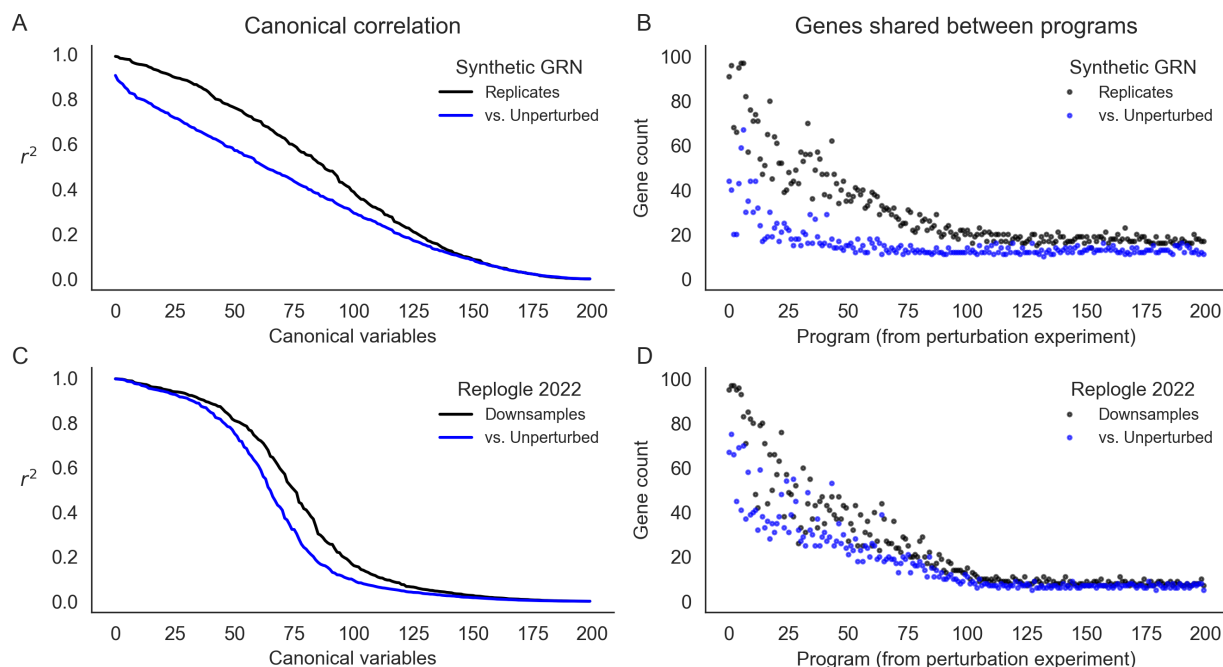
351 mance between coexpression and perturbation effects. Indeed, across networks in our study, we  
352 find that sparsity and modularity both tend to diminish the enrichment of true edges among pairs  
353 of genes with high co-expression (**Fig. S12**). The opposite is true for perturbation effects, which  
354 are better predictors of edges in these same networks – however, it is worth noting that for finding  
355 edges, perturbation effects outperformed coexpression summary statistics in every network (**Fig.**  
356 **S12**). Taken together, these results underscore the importance of perturbation data for inferring  
357 network edges, and may suggest limitations in the use of coexpression networks. However, nei-  
358 ther form of data is a panacea, and care is warranted in the analysis of real experimental data and  
359 in the development of structure learning algorithms.

## 360 2.7.2 Discovering group structure

361 Recent work has also attempted to identify trait-relevant sets of genes that act through coordinated  
362 effects in a particular cell type. These groups are sometimes called programs, and it is common to  
363 use dimensionality reduction techniques like singular value decomposition (SVD) or non-negative  
364 matrix factorization (NMF) on single-cell expression values to identify groups [8, 33]. In our ex-  
365 ample synthetic GRN and in the Perturb-seq data, we used a variant of this approach based on  
366 truncated SVD (TSVD) to assign genes to programs. As input, we used the set of 75,328 unperturbed  
367 cells from real data [9] and downsamples of the entire experiment to the same number  
368 of cells; for the synthetic data, we simulated the same number of cells from baseline or baseline  
369 and perturbed conditions, mimicking the composition of the real experiments. From these data,  
370 we computed the first 200 singular vectors of the expression data, using each vector to define a  
371 “program” of 100 genes with the largest loadings (**Methods**).

372 Here, we assess the extent to which these programs and their constituent singular vectors  
373 replicate across experiments from perturbed and unperturbed conditions. We use canonical cor-  
374 relation analysis (CCA) to assess the similarity of the singular vectors. This technique seeks to  
375 find rotations  $a, b$  of inputs  $x, y$  such that the correlation between  $a^\top x$  and  $b^\top y$  is maximized –  
376 the transformed inputs are called canonical variables, and we report their correlations when the  
377 inputs  $x, y$  are gene singular vectors from different experimental conditions (**Fig. 7A**). In the syn-  
378 thetic GRN, the canonical correlation steadily declines over the 200 dimensions of input. Notably,  
379 the magnitude of this correlation is similar when perturbation data are compared to a replicate or  
380 to unperturbed data. Even though this correlation is modest by the 100th set of canonical vari-  
381 ables, this trend suggests consistency between lower-dimensional representations of expression  
382 data regardless of cell state (perturbed or unperturbed).

383 At this sample size (75,238 cells) for the synthetic GRN, however, there is a difference in the re-  
384 producibility of individual programs across data sources (perturbed or unperturbed; **Fig. 7B**). For  
385 this, we compare programs computed from one perturbation experiment (the “reference”) to pro-  
386 grams from a replicate perturbation experiment and to programs from unperturbed cells. For each  
387 program from the reference, we report the maximum number of genes which overlap any other  
388 program computed from each of the other data sources. The first few programs (corresponding  
389 to the first few singular vectors) are highly reproducible in the replicate perturbation data. This  
390 overlap steadily declines to no effective overlap after the first  $\sim 100$  programs. Despite similar  
391 canonical correlation, the unperturbed data do not replicate the same programs to the same extent  
392 – there is modest overlap with the first few programs from the perturbation data, and this overlap



**Figure 7: Learned representations are similar between control and intervened-upon cells.** Concordance between low-rank representations of single cells in a simulated GRN (top row) and downsamples of experimental data (Replogle *et. al.*, 2022). **(A)** Correlation between the first 200 canonical variables (linear combinations of singular vectors) between samples of 75,328 baseline or baseline+perturbed cells from a synthetic GRN. **(B)** Overlap in gene programs inferred from singular value decomposition of single cell expression data. Programs are defined using singular vectors of gene expression from perturbed cells (x-axis), and intersected with programs analogously defined from baseline and additional perturbed cells. **(C)** Canonical correlation of control cells and two downsamples of the entire Perturb-seq experiment (Replogle *et. al.*, 2022). **(D)** Overlap in gene programs computed from control and downsamples of experimental Perturb-seq data.

393 decays very quickly (after the first  $\sim 20$  programs). When assessing these programs with respect  
 394 to the ( $k = 10$ ) ground truth groups of this network, we find that nearly all true groups are at  
 395 least modestly well represented by the top 50 programs from both data sources. However, the  
 396 programs from the perturbation data much better represent the true groups than those from the  
 397 unperturbed data (**Fig. S13**).

398 We find similar results in the genome-wide Perturb-seq data. Here, however, the first few  
 399 canonical variables are highly correlated, and the canonical correlation drops off precipitously be-  
 400 tween the 50th and 100th canonical variables (**Fig. 7C**). We also find that programs computed from  
 401 two downsamples of the entire experiment are about as reproducible as those from the simulated  
 402 data, but are slightly more similar to the programs from control cells (**Fig. 7D**). While this may  
 403 reflect some aspect of GRN structure, it is also related to the number of cells in the input data and  
 404 magnitude of intrinsic gene expression noise. Both tend to reduce canonical correlations and the  
 405 reproducibility of gene programs. In the real data, lowering the number of cells input to TSVD  
 406 lowers both canonical correlation and program similarity compared to the entire experiment; re-  
 407 latedly, we find that the 75,328 control cells exhibit comparable performance as 30,000 perturbed  
 408 cells at recovering representations from the full data (**Fig. S14**). In the synthetic networks, we find



409 that altering the level of global transcriptional noise ( $s$ ) alters the concordance between replicates  
410 and between perturbed and unperturbed cell states – at high levels of noise, there is little practical  
411 difference between programs derived from perturbed or unperturbed cells, but with low levels of  
412 noise, the programs from perturbation data are markedly different (**Fig. S15**). For presentation in  
413 **Fig. 7A-B**, we chose a level of noise ( $s = 0.3$ ) that recapitulated the qualitative behavior of the real  
414 data.

415 Taken together, these findings seem to suggest that the leading variance components of single-  
416 cell gene expression data will be similar across perturbed and unperturbed conditions, unless the  
417 magnitude of perturbation effects is greater than the level of intrinsic transcriptional noise. Re-  
418 latedly, we suspect that dimensionality reduction techniques will produce concordant represen-  
419 tations of both perturbed and unperturbed data under similar conditions, and that this similarity  
420 can propagate into gene sets derived from these representations. This begs a key line of ques-  
421 tions for future work: where, and how, do molecular perturbations add value in uncovering the  
422 sets of genes that are collectively important for cell-type and disease-specific processes? In light  
423 of the number of cells required for reliable inference at this scale, we anticipate that large atlas-  
424 style cell reference data (e.g., the Human Cell Atlas and similar resources [34–36]) should provide  
425 promising opportunities to reveal global aspects of network structure.

### 426 3 Discussion

427 In this work, we have presented a new model to simulate gene regulatory networks, with partic-  
428 ular emphasis on generating networks with realistic structural properties. We note that this  
429 algorithm may be of interest in contexts outside gene regulation – namely, in studies of scale-free  
430 networks with group-like structure. We also anticipate that our technique to simulate gene ex-  
431 pression from arbitrary networks may be useful for model development and benchmarking, or in  
432 other studies where network structures are known or may be hypothesized.

433 Here, we have highlighted the utility of our approach with simulations to develop intuition  
434 about key properties of GRNs, particularly in the context of molecular perturbations and coexpres-  
435 sion data. While our study design draws inspiration from recent works using Perturb-seq, we also  
436 acknowledge limitations to the realism of our model. In focusing on the equilibrium dynamics of  
437 cells of one type, we have ignored developmental trajectories and cell-type heterogeneity within  
438 tissues, both of which modify our assumptions about regulatory network structure. For the sake of  
439 computational efficiency in quantifying expression for thousands of genes, we have also eschewed  
440 detailed models of the biological synthesis and experimental measurement of cellular RNAs: in  
441 future work where it is critical to match distributions of count data from experiments, we encour-  
442 age modeling these complexities. Similar considerations may also be necessary for the application  
443 of our approach beyond modeling knockouts – for example, in studying genetic variation which  
444 affects gene expression.

445 Independently, our results suggest that the space of realistic network structures may be quite  
446 limited, and that it may be useful to consider this prior information in various inference settings.  
447 While our approach as outlined in this work is not optimized for inference, the algorithms we  
448 describe are generative, which means they could be used directly in applications for simulation-  
449 based inference. Although we used experimental data from K562 cells in this study, we anticipate

450 the high-level structural properties of GRNs will generalize across different cell types. Moreover,  
451 we observed through simulations that hallmark properties of GRNs tend to confer resilience to  
452 perturbations across multiple measures, reducing the number of sensitive target genes and large-  
453 effect master regulators. We do not suspect this is an incidental finding in light of the selection  
454 to which GRNs are subject over evolutionary time, and suspect that considering this type of con-  
455 straint may be insightful for future work.

456 Looking forward, we also anticipate that broad observational studies of diverse cell types and  
457 deep interventional studies of specific cell lines will both be useful in disentangling the basis of  
458 complex traits in regulatory networks. However, a key question remains in determining how best  
459 to leverage these data types towards a unifying understanding of cell biology. We suggest that  
460 a scaffolded approach to this problem may be useful. Where the scale of cell atlases presents a  
461 unique opportunity to learn transferable representations of cells across developmental states and  
462 tissues, perhaps including the discovery of cellular programs, these data are limited in their ability  
463 to resolve interactions between single genes. This is where perturbation data – however limited  
464 to specific cell types – retain critical value. Even as existing network inference algorithms expe-  
465 rience computational challenges in genome-scale applications, the modularity of GRNs suggests  
466 that piecewise inference strategies may be viable until these challenges are resolved. As efforts  
467 like these enhance our mechanistic understanding of biological networks, we hope that our work  
468 serves to provide general intuition on their salient structural properties. We are optimistic that  
469 understanding these principles will be useful for an array of challenges and highlight future op-  
470 portunities in functional genomics.

## 471 **4 Acknowledgements**

472 We would like to thank Mineto Ota, Emma Dann, Courtney Smith, Garyk Brixi, Josh Weinstock,  
473 and members of the Pritchard Lab at Stanford University for helpful comments and discussion  
474 related to this work. M.A. acknowledges support from a Microsoft Research PhD Fellowship,  
475 and from the National Library of Medicine (NLM) under training grant T15LM007033. This  
476 work was supported by the National Human Genome Research Institute (NHGRI) under grants  
477 R01HG008140 and U01HG012069 (J.K.P.) and by the National Institute of General Medical Sci-  
478 ences (NIGMS) under grant R01GM115889 (G.S.).

## 479 5 Methods

### 480 5.1 Graph generating parameters

#### 481 5.1.1 Sampling

482 A full description of the graph generating algorithm can be found in the main text, with the exact  
483 procedure given by **Algorithm 1**. Here, we provide additional intuition on its generating param-  
484 eters, and detail the scheme for sampling them.

485 In motivating our study, we highlight several key properties of gene regulatory networks:  
486 briefly, these are sparsity, modular groups, and asymmetric power-law degree distributions. In  
487 **Fig. 2** we show that these properties are individually tuned by parameters of our generating  
488 algorithm. When generating synthetic networks, we sample values for each parameter across one  
489 to three orders of magnitude. To cover these ranges, the values are spaced geometrically, and the  
490 extrema are chosen to overlap values which we believe to be consistent with biological intuition  
491 for a network of  $n = 2,000$  genes.

- 492 • Sparsity term  $p$ :  $\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}\}$ .
- 493 • Number of groups  $k$ :  $\{1, 5, 10, 50, 100\}$ .
- 494 • Modularity term  $w$ :  $\{1, 9, 40, 90, 400, 900\}$ .
- 495 • In-degree uniformity term  $\delta_{in}$ :  $\{10, 30, 100, 300\}$ .
- 496 • Out-degree uniformity term  $\delta_{out}$ :  $\{1, 3, 10, 30\}$ .

497 The sparsity term  $p$  is sampled so that the average number of regulators per gene spans from  
498 low single-digits to low double-digits. The number of groups is sampled from  $k = 100$ , which  
499 corresponds to a rough lower limit on the size of groups (20 genes), to  $k = 1$ , which corresponds to  
500 the dissolution of group structure and is equivalent to the algorithm from Bollobás *et. al.*, 2003 [28].  
501 The within-group affinity term  $w$ , which controls modularity, is sampled in a similar way:  $w = 1$   
502 also corresponds to the dissolution of group structure, again giving the algorithm from Bollobás  
503 *et. al.*, 2003, and  $w = 900$  gives an upper limit on modularity with respect to groups  $k$ . The in- and  
504 out-degree uniformity terms  $\delta_{in}, \delta_{out}$  are both sampled across orders of magnitude. The bounds for  
505 the in-degree term to be larger in magnitude, corresponding to the assumption that the in-degree  
506 distribution should be less dispersed (i.e., have fewer hubs) than the out-degree distribution, but  
507 the range of values is intentionally overlapping.

508 To produce the set of 1,920 GRNs used in the study, we simulated one network with every  
509 possible combination of parameters listed above: this totals  $4 \times 5 \times 6 \times 4 \times 4 = 1,920$  networks.

#### 510 5.1.2 Relationship to perturbation effects

511 We performed a regression analysis to estimate the effect of each graph generating parameter on  
512 the distribution of perturbation effects in the synthetic GRNs. Specifically, we regressed the logit-  
513 transformed fraction of genes in each GRN that are hub regulators or hub targets according to the

514 following equation:

$$\text{logit}(p_{\text{genes}}) \sim 1 + (1/p) + k' + w + \delta_{in} + \delta_{out},$$

515 where  $1/p$  is the inverse of the sparsity term,  $k'$  is a transformed number of groups (GRNs with  
516  $k = 1$  group are treated as GRNs with  $k = n = 2000$  groups; see **Fig. 4**), and  $w$ ,  $\delta_{in}$ , and  $\delta_{out}$  are as  
517 described above. The dependent variable of the regression,  $p_{\text{genes}}$  is either the fraction of genes in  
518 the GRN which are hub regulators or hub targets. These quantities are analyzed separately. Full  
519 results for each regression are in **Table S1** and **S2**.

## 520 **5.2 Expression simulation**

### 521 **5.2.1 Parameter selection**

522 An overview of our gene expression model can be found in the main text. Here, we describe  
523 the sampling strategy for the parameters of the model and give additional information on their  
524 interpretation. Recall that the expression,  $x_i$ , of gene  $i$  is influenced by the following variables:

- 525 • the baseline transcription rate,  $\alpha_i$ ,
- 526 • the degradation rate of RNAs,  $\ell_i$ ,
- 527 • effects from regulating genes,  $\beta_{ji}$ ,
- 528 • expression noise, with magnitude  $s$ .

529 Note that  $\alpha$  and  $\ell$  are properties of genes (nodes);  $\beta$  is a property of regulatory interactions (edges);  
530 and  $s$  is a global parameter for the entire network. During forward simulation from the discretized  
531 stochastic differential equation, we take steps of size  $\Delta t = 0.01$  as in prior work [31], and update  
532 expression values from  $x$  (at time  $t$ ) to  $x'$  (at time  $t + \Delta t$ ) according to the following:

$$x' = x + \Delta t \cdot (\sigma(\alpha + \beta x) - \ell x) + s\sqrt{\Delta t} \cdot x\mathcal{N}(0, I).$$

533 In the deterministic limit, this results in an equation satisfied by any potential steady-state

$$x^* = \sigma(\alpha + \beta x^*) / \ell,$$

534 where  $\sigma(x) = \frac{1}{1+e^{-x}}$  is the logistic sigmoid. When setting up the model given a graph structure  
535 from our generating algorithm, we simulate expression parameters according to the following  
536 scheme:

- 537 •  $\sigma(\alpha_i) \stackrel{\text{iid}}{\sim} \text{Beta}(2, 8)$ , under the assumption that genes have low but non-zero expression at  
538 baseline, in the absence of regulation – i.e.,  $\sigma(\alpha_i)$  is small. Here,  $\sigma(x) = \frac{1}{1+e^{-x}}$  is again the  
539 logistic sigmoid (expit) function.
- 540 •  $\ell_i \stackrel{\text{iid}}{\sim} \text{Beta}(8, 2)$ , under the assumption that the maximum expression of each gene,  $1/\ell_i$ , tends  
541 to be of a similar order of magnitude (close to one), but can vary. To prevent steady-state  
542 gene expression from being excessively large (by having small  $\ell_i$ ), we hard clip  $\ell_i$  to be at  
543 least as large as  $e^{-\alpha_i}$ .

- 544 •  $\beta_{ji} \stackrel{\text{iid}}{\sim} (2p_j - 1) \cdot (1 + \text{Half-Normal}(0, 1))$ , under the assumptions that regulatory interactions  
545 have a minimum strength ( $|\beta_{ji}| \geq 1$ ). Here,  $p_j \stackrel{\text{iid}}{\sim} \text{Bernoulli}(p = 0.8)$  is the probability that a  
546 regulator  $j$  acts as an activator rather than a repressor.
- 547 •  $s = 10^{-4}$ , fixed across all genes in the networks. This value is chosen to be as large as possible  
548 without limiting detection of very small KO effects. At this level of noise, we can reliably  
549 detect  $\log_2$  fold-changes down to the order of  $10^{-4}$  (**Fig. 3D**).

## 550 5.2.2 Forward simulation

551 Once parameters of the expression model are chosen for a specific GRN, we initialize the expres-  
552 sion of each gene  $x_i = 0$  and conduct forward simulation according to the update rule given in  
553 the previous section, which is also described in the main text.

554 When performing forward simulations, we initialize all genes in the network to have zero  
555 expression. We then perform  $b = 5,000$  iterations of forward simulation as a “burn-in”. After  
556 burn-in, we check whether the system of equations has converged to a steady-state by measuring  
557 differences in the time averaged mean after the burn-in. Specifically, we compute the maximum  
558 absolute  $\log_2$  fold-change of non-lowly expressed genes in the network

$$\max_{g, \bar{x}_{g,i} > s} \log_2 \left( \frac{\bar{x}_{g,i}}{\bar{x}_{g,i-h}} \right)$$

559 where  $g$  indexes genes whose running mean expression  $\bar{x}_{g,i}$  at the current iteration  $i$  is above the  
560 noise magnitude  $s$ , and  $h$  is the step size to check for convergence. Mathematically, the running  
561 mean in the numerator is

$$\bar{x}_{g,i} = \frac{1}{i-b} \sum_{t=b}^i x_{g,t}$$

562 where  $x_{g,t}$  is the expression of gene  $g$  at iteration  $t$ . The denominator  $\bar{x}_{g,i-h}$  is analogously the  
563 running mean expression of gene  $g$  the last time we checked for convergence,

$$\bar{x}_{g,i-h} = \frac{1}{i-h-b} \sum_{t=b}^{i-h} x_{g,t}.$$

564 If this maximum log fold-change is below  $10^{-3}$ , we say the system has converged, and take the  
565 vector  $\bar{x}_i$  as the steady-state expression of all genes in the network. We perform this check every  
566  $h = 1,000$  iterations, up to a maximum  $t_{max} = 20,000$  iterations. We take the vector  $\bar{x}_{t_{max}}$  as an  
567 approximate equilibrium state if the system did not fully converge.

568 We further assess the stability of the steady-state of each GRN by performing a linear stability  
569 analysis of the expression model in the limit  $s \rightarrow 0$ . In this limit, the expression model takes the  
570 form of an ordinary differential equation (ODE). The stability of an equilibrium point  $\bar{x}$  of this  
571 ODE can be assessed using the eigenvalues of the Jacobian matrix  $J$  evaluated at  $\bar{x}$  – if all of the  
572 eigenvalues have a negative real part, the system is said to be stable [37]. Here, we have

$$J = \left\{ \frac{\partial f(x_i)}{\partial x_j} \right\}_{ij}$$

573 where the  $(i, j)$ th entries correspond to the partial derivative of the deterministic part of the ex-  
574 pression function  $f(x_i)$  of gene  $i$ , with respect to the expression  $x_j$  of gene  $j$ . For our model,

$$\begin{aligned}\frac{\partial f(x_i)}{\partial x_j} &= \frac{\partial}{\partial x_j} (\sigma(\alpha_i + \sum_k \beta_{ki} x_k) - \ell_i x_i) \\ &= \beta_{ji} \sigma(\alpha_i + \sum_k \beta_{ki} x_k) [1 - \sigma(\alpha_i + \sum_k \beta_{ki} x_k)] - \mathbb{1}(i = j) \ell_i,\end{aligned}$$

575 where the first term is zero for  $\beta_{ji} = 0$  and the second term is zero for  $j \neq i$ .

### 576 5.2.3 Perturbation experiments

577 For each synthetic GRN in this study, we perform a systematic assessment of gene-level perturba-  
578 tion effects. We start with baseline steady-state expression values of an instantiated GRN, with  
579 edges drawn according to the generating algorithm and expression parameters chosen as de-  
580 scribed above. Then, separately for each gene  $j$ , we perform a knockout by setting  $\beta_{ji} = 0$  for  
581 all other genes  $i$  – that is, we nullify its outgoing effects. This perturbs the equilibrium dynamics  
582 of the expression SDE, and we conduct additional rounds of forward simulation using the mod-  
583 ified parameters until a new expression steady-state is reached. We perform the same procedure  
584 with burn-in and convergence checks as in the previous section.

585 We summarize the effect that perturbing gene  $j$  has on gene  $i$  using a log fold-change in ex-  
586 pression values:

$$\log_2 \text{FC}_{ji} = \log_2(x_i | \text{do}(x_j = 0)) - \log_2(x_i)$$

587 where  $x_i$  is the steady-state expression of gene  $i$  under baseline conditions, and  $x_i | \text{do}(x_j = 0)$  is its  
588 steady-state expression when gene  $j$  has been perturbed (both computed as described above).

### 589 5.2.4 Baseline coexpression

590 Since gene coexpression is also commonly used to describe pairwise relationships between genes,  
591 we further use the expression SDE to compute the gene-level correlations at steady-state in the  
592 synthetic GRNs. Under baseline conditions, we perform  $t = 10,000$  additional forward time steps,  
593 from which we sample  $m = 10,000$  “baseline cells” by taking the gene expression value at every  
594 step. The noise inherent to the model ( $s = 10^{-4}$ ) produces sufficient variability in this cell popu-  
595 lation to compute gene-level correlations. We measure the coexpression of genes all  $i$  and  $j$  (not  
596 filtering out lowly expressed genes) using the Pearson correlation between  $x_i$  and  $x_j$  across cells.

## 597 5.3 Perturb-seq data

### 598 5.3.1 Data processing

599 To motivate aspects of our work, and to assess our simulations in context of experimental data,  
600 we make use of summary statistics from a recent genome-scale Perturb-seq study [9]. Specifi-  
601 cally, we used pairwise FDR-corrected Anderson-Darling p-values (from the supplemental file,  
602 "anderson-darling p-values, BH-corrected.csv.gz") as a measure of the expression response  
603 to single-gene perturbations. Throughout this work, we used a single large subset of these data

604 corresponding to the set of genes whose expression was subject to both experimental perturba-  
605 tion and measurement in response. We matched perturbations to target genes using the provided  
606 ENSEMBL gene IDs, subsetting to perturbations which targeted any primary transcript. In (rare)  
607 cases where there was more than one such perturbation, we used the perturbation which induced  
608 a statistically significant change in the expression of the target transcript. We note that target genes  
609 with expression levels below 0.25 UMI per cell were not included in this file, which further limited  
610 the genes included in our analysis. We performed a similar post-processing step when analyzing  
611 results from the synthetic networks, limiting analysis to genes whose steady-state expression was  
612 above the level of intrinsic noise (i.e.,  $x_i > s$ ).

### 613 5.3.2 Comparing with simulations

614 We compared the distribution of perturbation effects (incoming and outgoing) at the gene level  
615 when assessing similarities between the real and simulated networks. For this, we thresholded  
616 pairwise effects from the experimental data at FDR-corrected Anderson-Darling  $p < 0.05$ , saying  
617 that effects at this significance level constitute biologically meaningful effects, and others do not.  
618 At this threshold, we find that 3.16% of pairwise effects are called significant. For a given gene  $i$ ,  
619 we then computed two values: the fraction of the network that is affected when  $i$  is perturbed (i.e.,  
620 the fraction of genes  $j$  for which  $p_{ij} < 0.05$ ), and the fraction of the network that affects  $i$  when  
621 perturbed (i.e., the fraction of genes  $j$  for which  $p_{ji} < 0.05$ ).

622 We then compared these distributions to analogous quantities derived from the synthetic  
623 GRNs. Since the experimental data are affected by imperfect statistical power, we set the dis-  
624 covery rate to be equal across all synthetic GRNs, doing so by taking the top 3.16% of pairwise  
625 perturbation effects (i.e.,  $|\log_2 \text{FC}|_{ji} > k$ , where  $k$  varies) as “statistically significant”. For each  
626 gene  $i$  in a synthetic GRN, we computed the fraction of the network which is affected when  $i$  is  
627 perturbed (i.e., the fraction of genes  $j$  for which  $|\log_2 \text{FC}|_{ij} > k$ ), and the fraction of the network  
628 which affects  $i$  when perturbed (i.e., the fraction of genes  $j$  for which  $|\log_2 \text{FC}|_{ji} > k$ ). Note that  
629 in each GRN in **Fig. 5**, we remove lowly-expressed genes, with baseline expression  $x_i < s$ . This  
630 means that the number of genes analyzed is not exactly the same for all GRNs – we therefore nor-  
631 malized the distribution of perturbation effects by the number of genes that are included in the  
632 analysis (i.e., those not lowly-expressed).

633 Finally, we compared the distributions of incoming and outgoing perturbation effects using  
634 the Kolmogorov-Smirnov test as implemented in `scipy` (`scipy.stats.ks_2samp`) [38]. This is a  
635 nonparametric test for equality of distribution between two samples, which measures the maxi-  
636 mum difference between cumulative distribution functions. To select the synthetic GRNs which  
637 are closest to the real data, we rank GRNs by largest KS  $p$ -values with each distribution (incom-  
638 ing and outgoing), then find the smallest rank  $r$  such that  $k$  GRNs are in the top  $r$  of all GRNs  
639 compared to both distributions.



## 640 5.4 Gene programs

### 641 5.4.1 Truncated singular value decomposition

642 We used truncated singular value decomposition (TSVD) to cluster genes into “programs” based  
643 on their expression profiles in cells from both perturbed and unperturbed settings, using the  
644 `TruncatedSVD` function from `scikit-learn` [39]. Briefly, TSVD is an algorithmic modification of  
645 singular value decomposition (SVD), which produces orthogonal singular vectors corresponding  
646 to the directions of maximum variance in the input data. In TSVD, only the top  $k$  vectors are  
647 computed, which results in faster computational runtimes for our analysis.

648 We assembled separate input datasets consisting of perturbed and unperturbed cells for both  
649 synthetic data and using downsamples of the experimental Perturb-seq data. For the synthetic  
650 data, we simulated 75,328 single cells from baseline conditions by forward simulation from the  
651 expression fixed point of the GRN, sampling cells from every forward time step. We also sim-  
652 ulated the expression of an identical number of cells under perturbed conditions, modeling the  
653 split of cells after the real Perturb-seq study: roughly 8.1% of the cells corresponded to baseline  
654 conditions, and the remainder were assigned uniformly at random to a knockout condition for  
655 each of the 2,000 genes in the GRN (this corresponds to 35 cells per KO on average). We do not  
656 filter out lowly expressed genes for this analysis.

657 For the real data, we used single-cell expression data of the 5,247 genes in our data subset  
658 from all 75,328 control cells as measurements of the GRN in unperturbed conditions. Then, to  
659 avoid effects from varying the size of the input cell population, we performed two independent  
660 (random) downsamples of the entire experiment to the same number of cells as measurements of  
661 the GRN in perturbed conditions.

662 With each of these input datasets  $X$ , we normalized each gene to have zero mean and unit vari-  
663 ance, and then performed TSVD to compute the top  $k = 200$  dimensions of expression variability.  
664 This resulted in singular matrices for cells ( $U$ ) and genes ( $V$ ), and a diagonal matrix of singular  
665 values,  $S$ . The product of these matrices approximates the input:

$$X \approx USV^T$$

666 and we used the gene loadings (columns  $v$  of the gene singular matrix  $V$ ) to define gene programs.  
667 Each “program” corresponds one-to-one with one of the gene singular vectors  $v$ , and is the set of  
668 100 genes with the largest squared entries of  $v$ .

### 669 5.4.2 Similarity across datasets

670 We assess the similarity of gene programs from two different experiments in two distinct ways:  
671 one using the set of genes which constitutes each program, and the other using the singular vector  
672 used to define it. When comparing programs  $\{p_i\}$  from one (reference) experiment to programs  
673  $\{p'_j\}$  from another experiment, we report the maximum overlap between each program  $p_i$  in the  
674 reference set to *any* program  $p'_j$  in other set – that is,

$$\text{overlap}(p_i, \{p'_j\}) = \max_j |p_i \cap p'_j|$$

675 which quantifies the extent to which each program is reproduced by the other experiment. When  
676 comparing gene singular vectors  $V = \{v_i\}$ ,  $V' = \{v'_j\}$  from the two experiments, we make use of  
677 the fact that the SVD of their dot product is a well-characterized mathematical procedure called  
678 canonical correlation analysis (CCA) [40]. The top  $k$  components of this decomposition are called  
679 canonical variables, and they each represent the axes of rotation which maximize correlation be-  
680 tween variables in the input data. Here, we report the canonical correlation (singular values from  
681 the second SVD step) for the first 200 canonical variables, to quantify the extent to which the  
682 lower-dimensional representations of the inputs are consistent with one another.

## 683 References

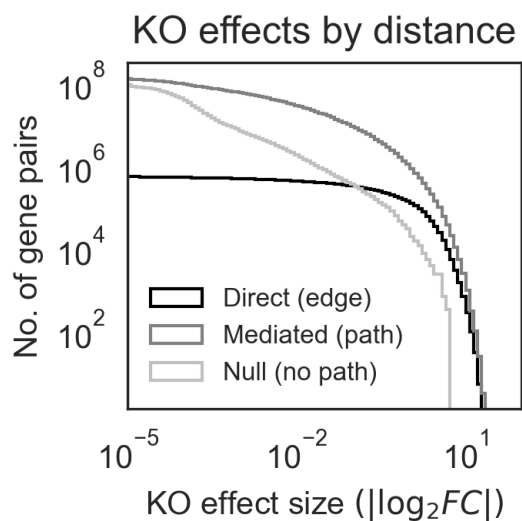
- 684 [1] Efthymia Papalexi and Rahul Satija. Single-cell RNA sequencing to explore immune cell  
685 heterogeneity. *Nature Reviews Immunology*, 18(1):35–45, January 2018. Publisher: Nature Pub-  
686 lishing Group.
- 687 [2] Gioele La Manno, Ruslan Soldatov, Amit Zeisel, Emelie Braun, Hannah Hochgerner, Viktor  
688 Petukhov, Katja Lidschreiber, Maria E. Kastrioti, Peter Lönnerberg, Alessandro Furlan, Jean  
689 Fan, Lars E. Borm, Zehua Liu, David van Bruggen, Jimin Guo, Xiaoling He, Roger Barker,  
690 Erik Sundström, Gonçalo Castelo-Branco, Patrick Cramer, Igor Adameyko, Sten Linnarsson,  
691 and Peter V. Kharchenko. RNA velocity of single cells. *Nature*, 560(7719):494–498, August  
692 2018. Publisher: Nature Publishing Group.
- 693 [3] Volker Bergen, Marius Lange, Stefan Peidli, F. Alexander Wolf, and Fabian J. Theis. General-  
694 izing RNA velocity to transient cell states through dynamical modeling. *Nature Biotechnology*,  
695 38(12):1408–1414, December 2020. Publisher: Nature Publishing Group.
- 696 [4] Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P. Fulco, Livnat Jerby-Arnon, Ne-  
697 manja D. Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychowdhury, Britt Adamson,  
698 Thomas M. Norman, Eric S. Lander, Jonathan S. Weissman, Nir Friedman, and Aviv Regev.  
699 Perturb-Seq: Dissecting Molecular Circuits with Scalable Single-Cell RNA Profiling of Pooled  
700 Genetic Screens. *Cell*, 167(7):1853–1866.e17, December 2016.
- 701 [5] Joseph M. Replogle, Thomas M. Norman, Albert Xu, Jeffrey A. Hussmann, Jin Chen, J. Zach-  
702 ery Cogan, Elliott J. Meer, Jessica M. Terry, Daniel P. Riordan, Niranjana Srinivas, Ian T. Fiddes,  
703 Joseph G. Arthur, Luigi J. Alvarado, Katherine A. Pfeiffer, Tarjei S. Mikkelsen, Jonathan S.  
704 Weissman, and Britt Adamson. Combinatorial single-cell CRISPR screens by direct guide  
705 RNA capture and targeted sequencing. *Nature Biotechnology*, 38(8):954–961, August 2020.  
706 Publisher: Nature Publishing Group.
- 707 [6] Jacob W. Freimer, Oren Shaked, Sahin Naqvi, Nasa Sinnott-Armstrong, Arwa Kathiria, Chris-  
708 tian M. Garrido, Amy F. Chen, Jessica T. Cortez, William J. Greenleaf, Jonathan K. Pritchard,  
709 and Alexander Marson. Systematic discovery and perturbation of regulatory genes in human  
710 T cells reveals the architecture of immune networks. *Nature Genetics*, 54(8):1133–1144, August  
711 2022. Number: 8 Publisher: Nature Publishing Group.
- 712 [7] Joshua S. Weinstock, Maya M. Arce, Jacob W. Freimer, Mineto Ota, Alexander Marson, Alexis  
713 Battle, and Jonathan K. Pritchard. Gene regulatory network inference from CRISPR pertur-  
714 bations in primary CD4+ T cells elucidates the genomic basis of immune disease. *bioRxiv*,  
715 page 2023.09.17.557749, October 2023.
- 716 [8] Gavin R. Schnitzler, Helen Kang, Shi Fang, Ramcharan S. Angom, Vivian S. Lee-Kim, X. Rosa  
717 Ma, Ronghao Zhou, Tony Zeng, Katherine Guo, Martin S. Taylor, Shamsudheen K. Vel-  
718 larikkal, Aurelie E. Barry, Oscar Sias-Garcia, Alex Bloemendal, Glen Munson, Philine Guckel-  
719 berger, Tung H. Nguyen, Drew T. Bergman, Stephen Hinshaw, Nathan Cheng, Brian Cleary,  
720 Krishna Aragam, Eric S. Lander, Hilary K. Finucane, Debabrata Mukhopadhyay, Rajat M.  
721 Gupta, and Jesse M. Engreitz. Convergence of coronary artery disease genes onto endothelial  
722 cell programs. *Nature*, 626(8000):799–807, February 2024. Number: 8000 Publisher: Nature  
723 Publishing Group.

- 724 [9] Joseph M. Replogle, Reuben A. Saunders, Angela N. Pogson, Jeffrey A. Hussmann, Alexan-  
725 der Lenail, Alina Guna, Lauren Mascibroda, Eric J. Wagner, Karen Adelman, Gila Lithwick-  
726 Yanai, Nika Iremadze, Florian Oberstrass, Doron Lipson, Jessica L. Bonnar, Marco Jost,  
727 Thomas M. Norman, and Jonathan S. Weissman. Mapping information-rich genotype-  
728 phenotype landscapes with genome-scale Perturb-seq. *Cell*, 185(14):2559–2575.e28, July 2022.
- 729 [10] Mathieu Chevalley, Jacob Sackett-Sanders, Yusuf Roohani, Pascal Notin, Artemy Bakulin,  
730 Dariusz Brzezinski, Kaiwen Deng, Yuanfang Guan, Justin Hong, Michael Ibrahim, Wojciech  
731 Kotlowski, Marcin Kowiel, Panagiotis Misiakos, Achille Nazaret, Markus Püschel, Chris  
732 Wendler, Arash Mehrjou, and Patrick Schwab. The CausalBench challenge: A machine  
733 learning contest for gene network inference from single-cell perturbation data, August 2023.  
734 arXiv:2308.15395 [cs, q-bio].
- 735 [11] Mathieu Chevalley, Yusuf Roohani, Arash Mehrjou, Jure Leskovec, and Patrick Schwab.  
736 CausalBench: A Large-scale Benchmark for Network Inference from Single-cell Perturbation  
737 Data, July 2023. arXiv:2210.17283 [cs].
- 738 [12] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe’er. Using Bayesian networks to  
739 analyze expression data. In *Proceedings of the fourth annual international conference on Com-  
740 putational molecular biology, RECOMB ’00*, pages 127–135, New York, NY, USA, April 2000.  
741 Association for Computing Machinery.
- 742 [13] David Heckerman. A Tutorial on Learning with Bayesian Networks. In Dawn E. Holmes  
743 and Lakhmi C. Jain, editors, *Innovations in Bayesian Networks: Theory and Applications*, pages  
744 33–82. Springer, Berlin, Heidelberg, 2008.
- 745 [14] Yuhao Wang, Liam Solus, Karren Dai Yang, and Caroline Uhler. Permutation-based Causal  
746 Inference Algorithms with Interventions, November 2017. arXiv:1705.10220 [stat].
- 747 [15] Romain Lopez, Jan-Christian Hütter, Jonathan K. Pritchard, and Aviv Regev. Large-Scale  
748 Differentiable Causal Discovery of Factor Graphs, October 2022. arXiv:2206.07824 [cs, q-bio,  
749 stat].
- 750 [16] Douglas Yao, Loic Binan, Jon Bezney, Brooke Simonton, Jahanara Freedman, Chris J. Frang-  
751 ieh, Kushal Dey, Kathryn Geiger-Schuller, Basak Eraslan, Alexander Gusev, Aviv Regev, and  
752 Brian Cleary. Scalable genetic screening for regulatory circuits using compressed Perturb-seq.  
753 *Nature Biotechnology*, pages 1–14, October 2023. Publisher: Nature Publishing Group.
- 754 [17] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network Mo-  
755 tifs: Simple Building Blocks of Complex Networks. *Science*, 298(5594):824–827, October 2002.  
756 Publisher: American Association for the Advancement of Science.
- 757 [18] Shai S. Shen-Orr, Ron Milo, Shmoolik Mangan, and Uri Alon. Network motifs in the tran-  
758 scriptional regulation network of *Escherichia coli*. *Nature Genetics*, 31(1):64–68, May 2002.  
759 Publisher: Nature Publishing Group.
- 760 [19] C. Seshadhri, Aneesh Sharma, Andrew Stolman, and Ashish Goel. The impossibility of low-  
761 rank representations for triangle-rich complex networks. *Proceedings of the National Academy  
762 of Sciences*, 117(11):5631–5637, March 2020. Publisher: Proceedings of the National Academy  
763 of Sciences.

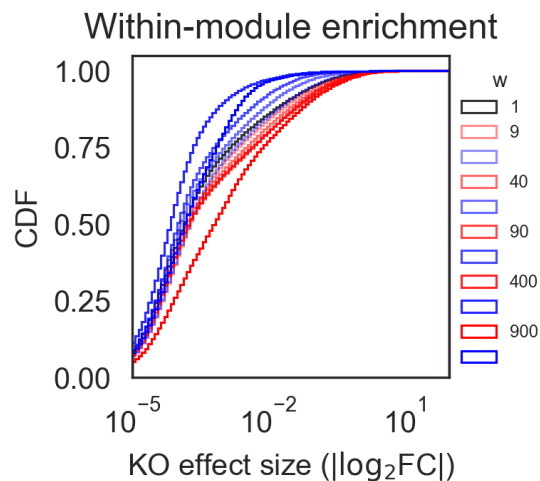
- 764 [20] S. Balaji, Lakshminarayan M. Iyer, L. Aravind, and M. Madan Babu. Uncovering a Hidden  
765 Distributed Architecture Behind Scale-free Transcriptional Regulatory Networks. *Journal of*  
766 *Molecular Biology*, 360(1):204–212, June 2006.
- 767 [21] Mark B. Gerstein, Anshul Kundaje, Manoj Hariharan, Stephen G. Landt, Koon-Kiu Yan, Chao  
768 Cheng, Xinmeng Jasmine Mu, Ekta Khurana, Joel Rozowsky, Roger Alexander, Renqiang  
769 Min, Pedro Alves, Alexej Abyzov, Nick Addleman, Nitin Bhardwaj, Alan P. Boyle, Philip  
770 Cayting, Alexandra Charos, David Z. Chen, Yong Cheng, Declan Clarke, Catharine Eastman,  
771 Ghia Euskirchen, Seth Fretze, Yao Fu, Jason Gertz, Fabian Grubert, Arif Harmanci, Preti Jain,  
772 Maya Kasowski, Phil Lacroute, Jing Leng, Jin Lian, Hannah Monahan, Henriette O’Geen,  
773 Zhengqing Ouyang, E. Christopher Partridge, Dorrelyn Patacsil, Florencia Pauli, Debasish  
774 Raha, Lucia Ramirez, Timothy E. Reddy, Brian Reed, Minyi Shi, Teri Slifer, Jing Wang, Linfeng  
775 Wu, Xinqiong Yang, Kevin Y. Yip, Gili Zilberman-Schapira, Serafim Batzoglou, Arend Sidow,  
776 Peggy J. Farnham, Richard M. Myers, Sherman M. Weissman, and Michael Snyder. Architec-  
777 ture of the human regulatory network derived from ENCODE data. *Nature*, 489(7414):91–100,  
778 September 2012. Publisher: Nature Publishing Group.
- 779 [22] Arun J. Singh, Stephen A. Ramsey, Theresa M. Filtz, and Chrissa Kioussi. Differential  
780 gene regulatory networks in development and disease. *Cellular and Molecular Life Sciences*,  
781 75(6):1013–1025, March 2018.
- 782 [23] Peter Langfelder and Steve Horvath. WGCNA: an R package for weighted correlation net-  
783 work analysis. *BMC Bioinformatics*, 9(1):559, December 2008.
- 784 [24] Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmod-  
785 els: First steps. *Social Networks*, 5(2):109–137, June 1983.
- 786 [25] Thorben Funke and Till Becker. Stochastic block models: A comparison of variants and infer-  
787 ence methods. *PLOS ONE*, 14(4):e0215296, April 2019. Publisher: Public Library of Science.
- 788 [26] Albert-László Barabási and Réka Albert. Emergence of Scaling in Random Networks. *Science*,  
789 286(5439):509–512, October 1999. Publisher: American Association for the Advancement of  
790 Science.
- 791 [27] Petter Holme and Beom Jun Kim. Growing scale-free networks with tunable clustering. *Physi-  
792 cal Review E*, 65(2):026107, January 2002. Publisher: American Physical Society.
- 793 [28] Béla Bollobás, Christian Borgs, Jennifer T Chayes, and Oliver Riordan. Directed scale-free  
794 graphs. In *SODA*, volume 3, pages 132–139, 2003.
- 795 [29] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of ‘small-world’ networks.  
796 *Nature*, 393(6684):440–442, June 1998. Publisher: Nature Publishing Group.
- 797 [30] Reuven Cohen and Shlomo Havlin. Scale-Free Networks Are Ultrasmall. *Physical Review  
798 Letters*, 90(5):058701, February 2003. Publisher: American Physical Society.
- 799 [31] Payam Dibaeinia and Saurabh Sinha. SERGIO: A Single-Cell Expression Simulator Guided  
800 by Gene Regulatory Networks. *Cell Systems*, 11(3):252–271.e11, September 2020.
- 801 [32] Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. DAGs with NO TEARS:  
802 Continuous Optimization for Structure Learning, November 2018. arXiv:1803.01422 [cs, stat].

- 803 [33] Dylan Kotliar, Adrian Veres, M Aurel Nagy, Shervin Tabrizi, Eran Hodis, Douglas A Melton,  
804 and Pardis C Sabeti. Identifying gene expression programs of cell-type identity and cellular  
805 activity with single-cell RNA-Seq. *eLife*, 8:e43803, July 2019. Publisher: eLife Sciences  
806 Publications, Ltd.
- 807 [34] Aviv Regev, Sarah A Teichmann, Eric S Lander, Ido Amit, Christophe Benoist, Ewan Birney,  
808 Bernd Bodenmiller, Peter Campbell, Piero Carninci, Menna Clatworthy, Hans Clevers, Bart  
809 Deplancke, Ian Dunham, James Eberwine, Roland Eils, Wolfgang Enard, Andrew Farmer,  
810 Lars Fugger, Berthold Göttgens, Nir Hacohen, Muzlifah Haniffa, Martin Hemberg, Seung  
811 Kim, Paul Klenerman, Arnold Kriegstein, Ed Lein, Sten Linnarsson, Emma Lundberg, Joakim  
812 Lundeberg, Partha Majumder, John C Marioni, Miriam Merad, Musa Mhlanga, Martijn Naw-  
813 ijn, Mihai Netea, Garry Nolan, Dana Pe'er, Anthony Phillipakis, Chris P Ponting, Stephen  
814 Quake, Wolf Reik, Orit Rozenblatt-Rosen, Joshua Sanes, Rahul Satija, Ton N Schumacher,  
815 Alex Shalek, Ehud Shapiro, Padmanee Sharma, Jay W Shin, Oliver Stegle, Michael Stratton,  
816 Michael J T Stubbington, Fabian J Theis, Matthias Uhlen, Alexander van Oudenaarden, Al-  
817 lon Wagner, Fiona Watt, Jonathan Weissman, Barbara Wold, Ramnik Xavier, Nir Yosef, and  
818 Human Cell Atlas Meeting Participants. The Human Cell Atlas. *eLife*, 6:e27041, December  
819 2017. Publisher: eLife Sciences Publications, Ltd.
- 820 [35] THE TABULA SAPIENS CONSORTIUM. The Tabula Sapiens: A multiple-organ, single-cell  
821 transcriptomic atlas of humans. *Science*, 376(6594):eabl4896, May 2022. Publisher: American  
822 Association for the Advancement of Science.
- 823 [36] Colin Megill, Bruce Martin, Charlotte Weaver, Sidney Bell, Lia Prins, Seve Badajoz, Brian Mc-  
824 Candless, Angela Oliveira Pisco, Marcus Kinsella, Fiona Griffin, Justin Kiggins, Genevieve  
825 Haliburton, Arathi Mani, Matthew Weiden, Madison Dunitz, Maximilian Lombardo, Timmy  
826 Huang, Trent Smith, Signe Chambers, Jeremy Freeman, Jonah Cool, and Ambrose Carr. cel-  
827 lXgene: a performant, scalable exploration platform for high dimensional sparse matrices,  
828 April 2021. Pages: 2021.04.05.438318 Section: New Results.
- 829 [37] Steven H Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry,*  
830 *and engineering*. CRC press, 2018.
- 831 [38] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David  
832 Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J.  
833 van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, An-  
834 drew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng,  
835 Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Hen-  
836 riksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian  
837 Pedregosa, and Paul van Mulbregt. SciPy 1.0: fundamental algorithms for scientific com-  
838 puting in Python. *Nature Methods*, 17(3):261–272, March 2020. Publisher: Nature Publishing  
839 Group.
- 840 [39] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion,  
841 Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-  
842 learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.
- 843 [40] William H Press. Canonical correlation clarified by singular value decomposition, 2011.

## 844 Supplementary Information

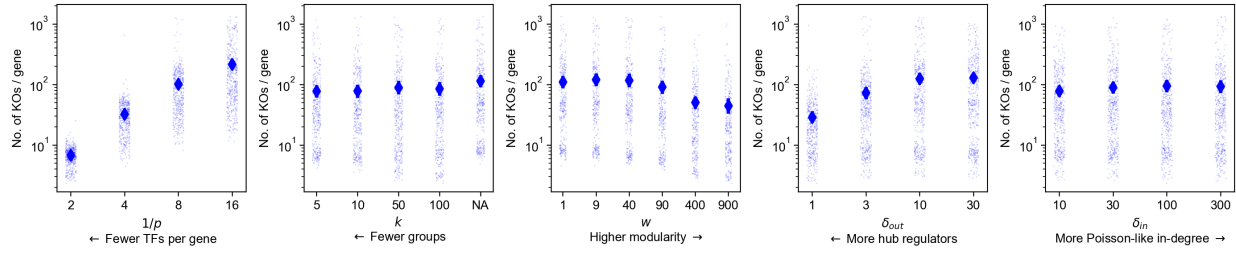


**Figure S1: Mediated effects outnumber direct effects at most magnitudes.** Same as Fig. 3D, but with distances binned by whether pairs of genes are connected by an edge (distance 1, a “direct effect”), any path (distance greater than 1, a “mediated effect”), or no path at all (“null”). Note also that the  $y$ -axis is the count of gene pairs with a perturbation effect of at least the magnitude given on the  $x$ -axis – that is, the distribution shown is a non-normalized inverse CDF. Gene pairs are pooled from the 50 example GRNs in Fig. 3.

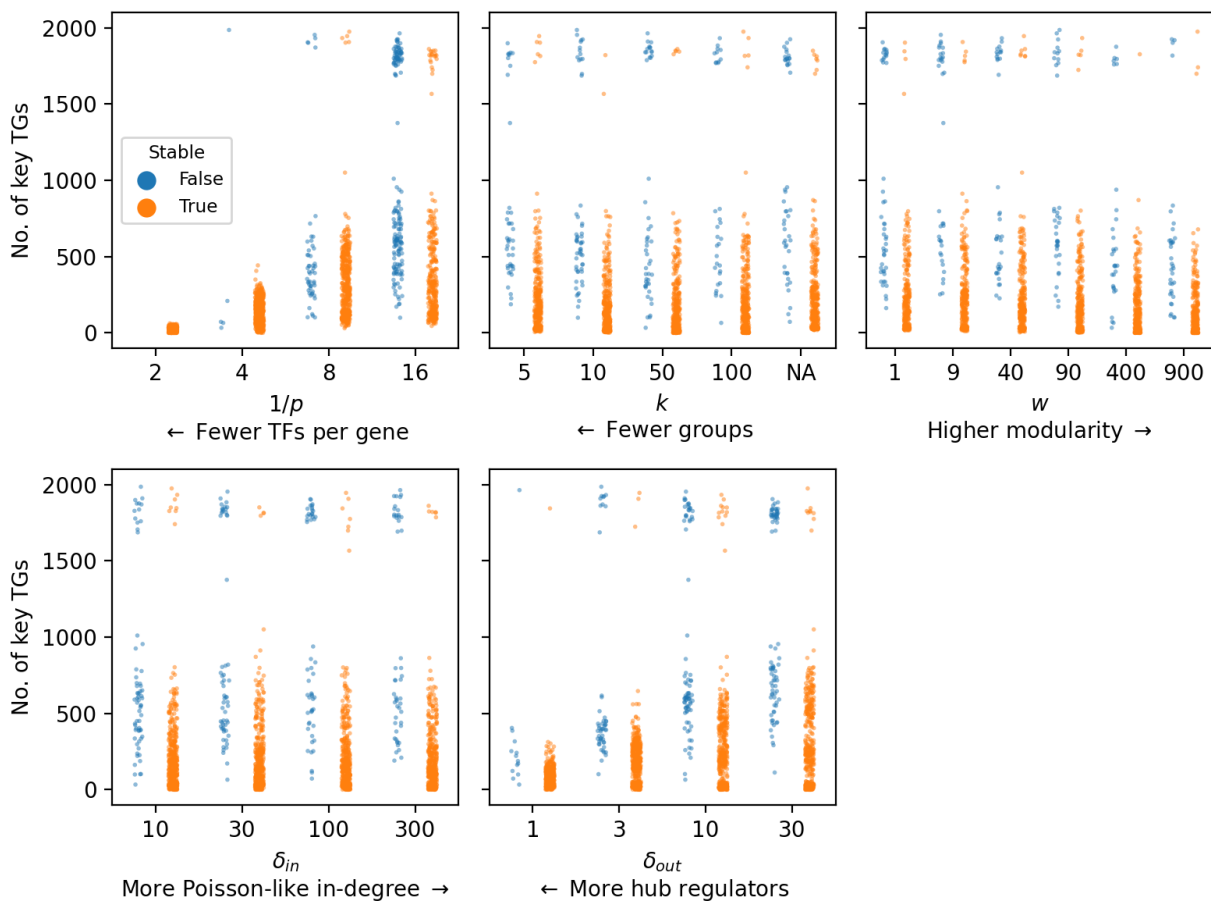


**Figure S2: Modularity term differentiates within- and between-module effects.** Same as Fig. 3E, with within-module perturbation effects in red and between-module perturbation effects in blue. Here, networks are chosen so as to highlight the effect of the modularity term  $w$ . Each pair of blue and red tracelines is distribution of the within- (red) or across-module perturbation effects a single GRN. The generating parameters for these GRNs vary  $w$  (see legend) but hold other parameters constant, as follow:  $p = 1/4$ ,  $k = 50$ ,  $\delta_{in} = 10$ ,  $\delta_{out} = 10$ .

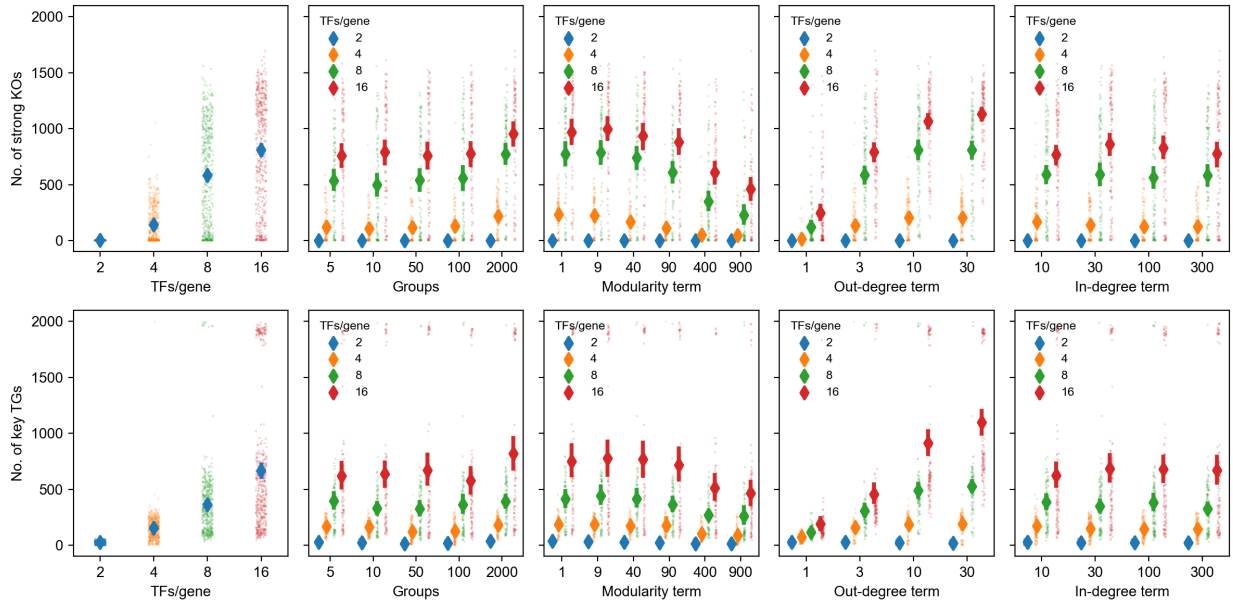




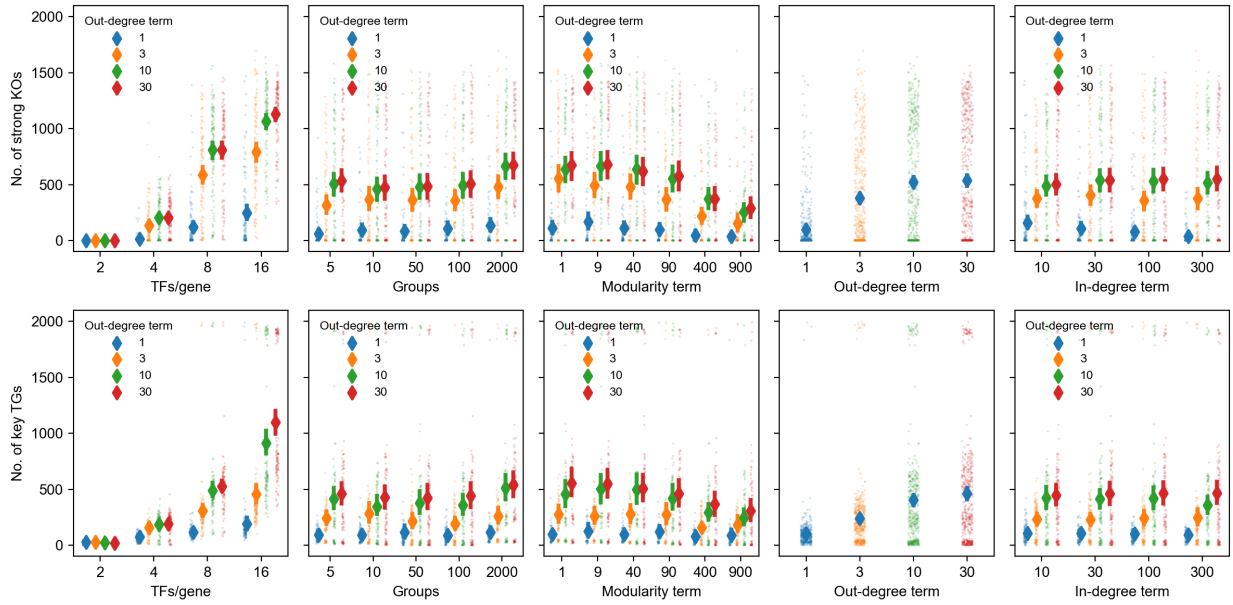
**Figure S3: Network generating parameters affect the number of KO effects.** Same as Fig. 4, but with summary statistic ( $y$ -axis) as the average number of perturbation effects per gene in the GRN with  $|\log_2 FC| \geq 0.1$ . We observe a similar direction of effect for each parameter as with the statistics presented in the main text.



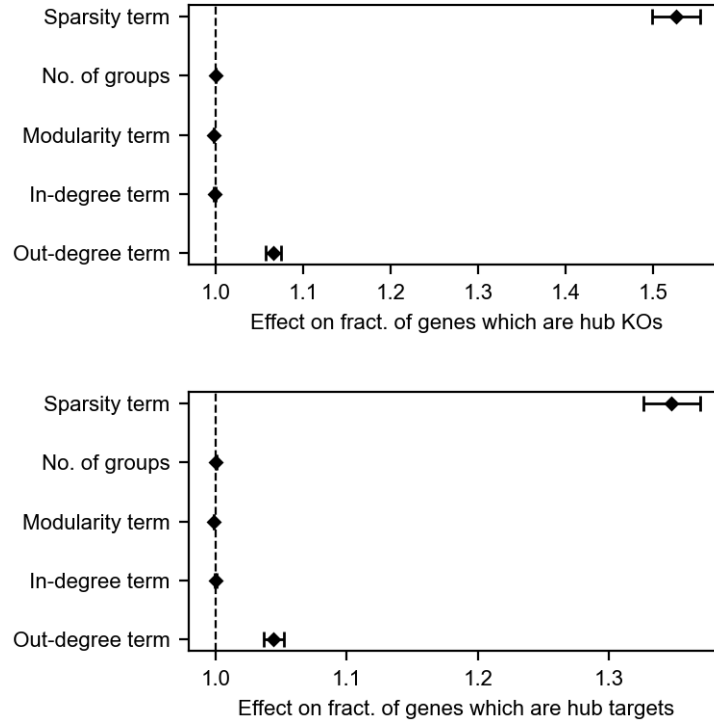
**Figure S4: Network generating parameters affect the stability of the fixed point.** Same as Fig. 4, only showing the number of key target genes and stratifying by whether the expression equilibrium point of the synthetic GRN is stable (**Methods**). In all, 1,693 of the 1,920 GRNs (88.2%) reach an expression equilibrium through forward simulation of the SDE which is a stable fixed point of the corresponding ODE. These GRNs tend to be sparse (lower  $1/p$ ), modular (higher  $w$ ), and have more hub regulators (lower  $\delta_{out}$ ).



**Figure S5: No interaction between sparsity term and other network generating parameters.** Same as Fig. 4, but with additional stratification by the sparsity term  $1/p$ . There is no obvious visual evidence for interactions between the parameters.



**Figure S6: No interaction between out-degree term and other network generating parameters.** Same as Fig. 4, but with additional stratification by the out-degree term  $\delta_{out}$ . There is no obvious visual evidence for interactions between the parameters.



**Figure S7: Summary of regression models for effects of network parameters on perturbations.** Coefficients from regressing the logit-transformed fraction of genes which are hub knockouts (top) or target genes (bottom) on network generating parameters. Errorbars denote 95% confidence intervals for the regression coefficients. Model summaries can be found in Tables S1 and S2.

<b>Dep. Variable:</b>	logit(pct_ko)	<b>R-squared:</b>	0.587
<b>Model:</b>	OLS	<b>Adj. R-squared:</b>	0.586
<b>Method:</b>	Least Squares	<b>F-statistic:</b>	543.6
<b>Date:</b>	–	<b>Prob (F-statistic):</b>	0.00
<b>Time:</b>	–	<b>Log-Likelihood:</b>	-4167.4
<b>No. Observations:</b>	1920	<b>AIC:</b>	8347.
<b>Df Residuals:</b>	1914	<b>BIC:</b>	8380.
<b>Df Model:</b>	5		
<b>Covariance Type:</b>	nonrobust		

	coef	std err	t	P>  t	[0.025	0.975]
<b>const</b>	-6.9629	0.115	-60.458	0.000	-7.189	-6.737
<b>r</b>	0.4229	0.009	46.789	0.000	0.405	0.441
<b>k_adj</b>	0.0004	6.18e-05	6.857	0.000	0.000	0.001
<b>w</b>	-0.0023	0.000	-15.724	0.000	-0.003	-0.002
<b>delta_in</b>	-0.0012	0.000	-2.801	0.005	-0.002	-0.000
<b>delta_out</b>	0.0637	0.004	15.061	0.000	0.055	0.072

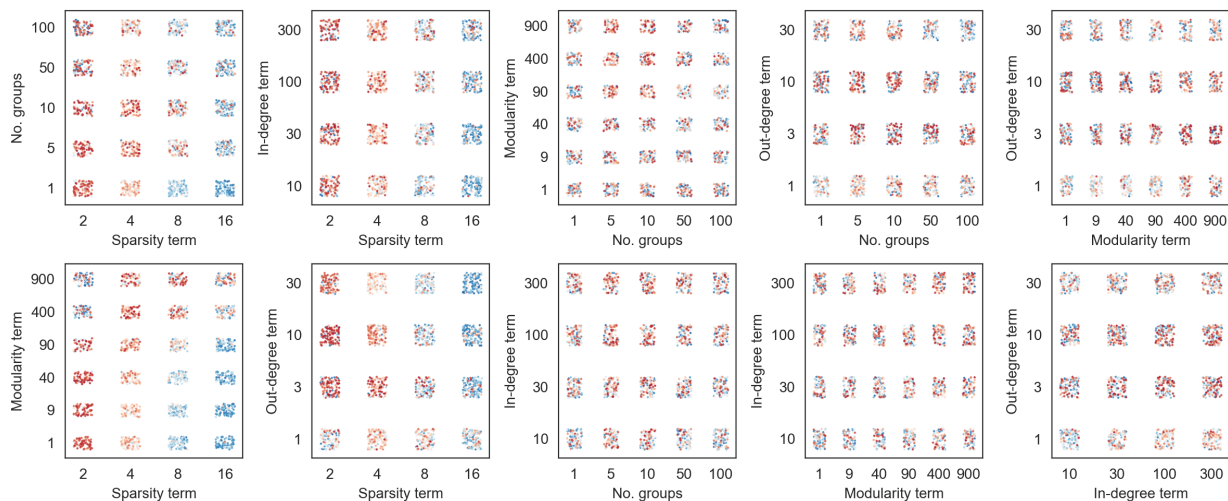
**Table S1:** Summary of regression results (fraction of genes which are hub knockouts).

<b>Dep. Variable:</b>	logit(pct_tg)	<b>R-squared:</b>	0.461
<b>Model:</b>	OLS	<b>Adj. R-squared:</b>	0.460
<b>Method:</b>	Least Squares	<b>F-statistic:</b>	327.7
<b>Date:</b>	–	<b>Prob (F-statistic):</b>	6.30e-254
<b>Time:</b>	–	<b>Log-Likelihood:</b>	-3973.5
<b>No. Observations:</b>	1920	<b>AIC:</b>	7959.
<b>Df Residuals:</b>	1914	<b>BIC:</b>	7992.
<b>Df Model:</b>	5		
<b>Covariance Type:</b>	nonrobust		

	coef	std err	t	P>  t	[0.025	0.975]
<b>const</b>	-4.7854	0.104	-45.966	0.000	-4.990	-4.581
<b>r</b>	0.2983	0.008	36.514	0.000	0.282	0.314
<b>k_adj</b>	0.0003	5.59e-05	5.686	0.000	0.000	0.000
<b>w</b>	-0.0016	0.000	-12.002	0.000	-0.002	-0.001
<b>delta_in</b>	-0.0002	0.000	-0.438	0.661	-0.001	0.001
<b>delta_out</b>	0.0433	0.004	11.345	0.000	0.036	0.051

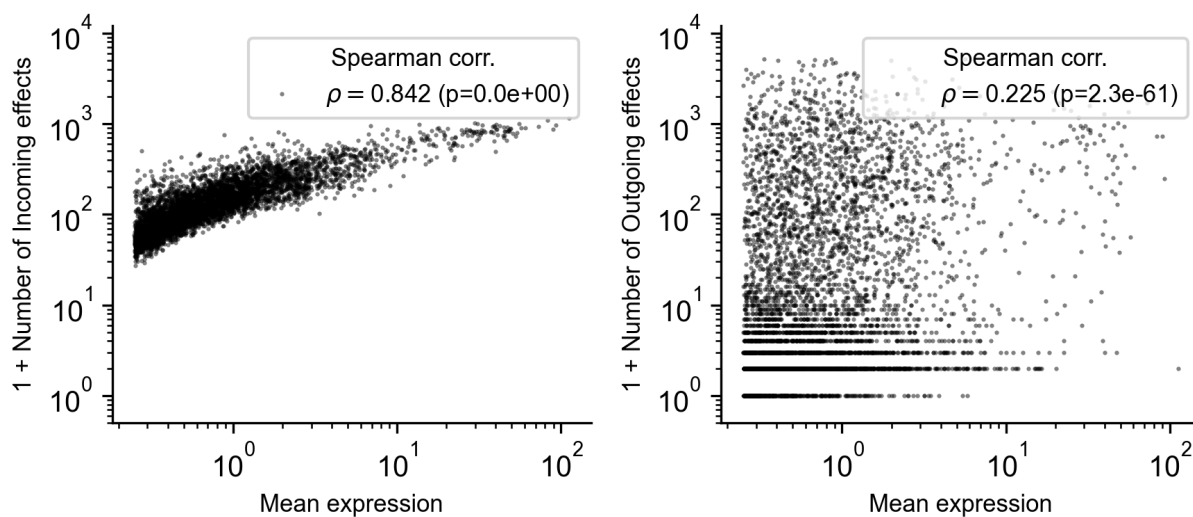
**Table S2:** Summary of regression results (fraction of genes which are hub target genes).



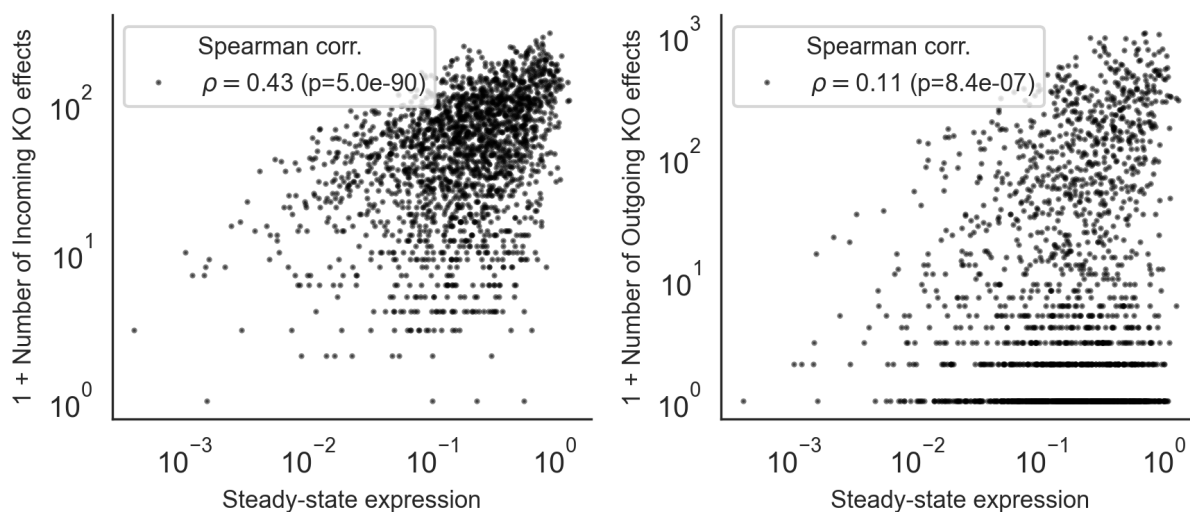
**Figure S8: No interaction between network generating parameters and fit to experimental data.** As in Fig. 5C-E, we show the relationship between pairs of network generating parameters and goodness of fit to the cumulative distribution of perturbation effects from experimental Perturb-seq data. Each GRN (one point in every subpanel) is colored by its ranked fit to data: the synthetic GRNs are ranked separately by Kolmogorov-Smirnov  $p$ -value for incoming and outgoing perturbation effects, then the sum of these two ranks is used to produce an overall ranking. Intense red color indicates better ranked fit to data, and intense blue color indicates a worse ranking.



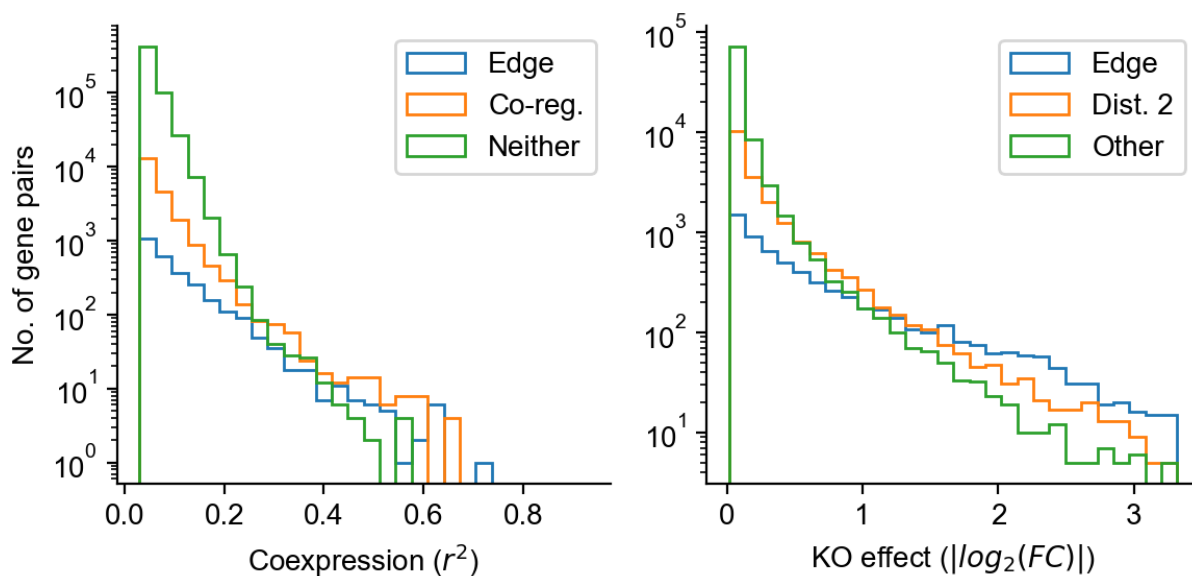
### Replogle 2022 (GWPS)



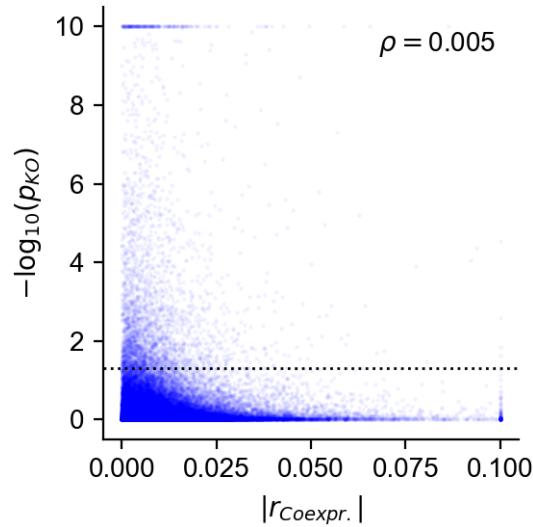
### Well-matched (synthetic) GRN



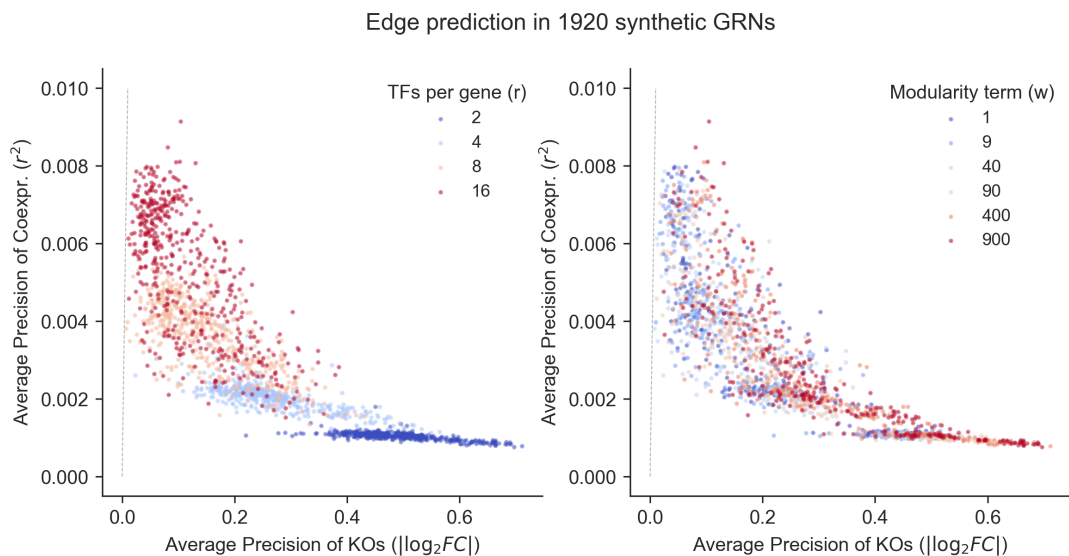
**Figure S9: Baseline expression influences the number of outgoing and incoming perturbation effects.** In the subsetted data from Replogle et. al. [9], and in the focal GRN from Fig. 6, we show the relationship between mean expression (in control cells in the experimental data; top panels) or steady-state expression (in the synthetic GRN; bottom panels) and the number of incoming (left) or outgoing (right) perturbation effects. Baseline expression relates to both of these quantities in both data sets: the relationships are stronger in the experimental data in part due to limits on detection power (especially important for incoming effects).



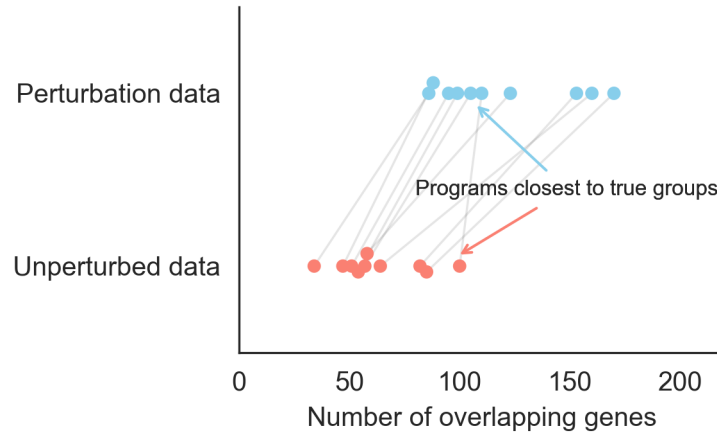
**Figure S10: Coexpression is more often due to coregulation than edges.** In the focal GRN from Fig. 6, we show a histogram of coexpression values split by whether pairs of genes share an edge (“A regulates B, or B regulates A”, share a regulator (“A and B are coregulated”), or have another relationship (left panel). Similarly, for perturbation effects, we show the distribution split by whether pairs of genes share an edge (“A regulates B”), a path of distance 2 (“A indirectly regulates B”), or another relationship (right panel). At nearly all levels of coexpression, coregulation is more common than direct regulation. Meanwhile, direct regulation is more common than indirect regulation for the largest perturbation effects – note that the range of KO effects is clipped as in Fig. 6.



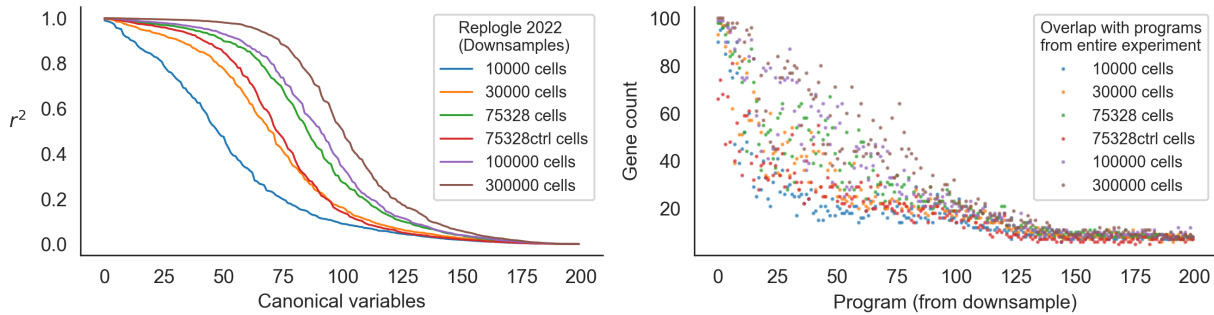
**Figure S11: Baseline coexpression and perturbation effects are uncorrelated in Perturb-seq data.** Same as **Fig. 6E**, using data from our analysis subset of Replogle *et. al.* 2022 [9]. Gene co-expression (x-axis) is the unsigned Pearson correlation between normalized single-cell gene expression data from unperturbed cells (clipped at  $|r| = 0.1$ ). Perturbation effects (y-axis) are pairwise log-transformed Anderson-Darling p-values for differences in gene expression distribution between perturbed and unperturbed states (clipped at  $-\log_{10}(p) = 10$ ). Rank correlation (Spearman's  $\rho$ ) is computed on the transformed but not clipped values of these two statistics.



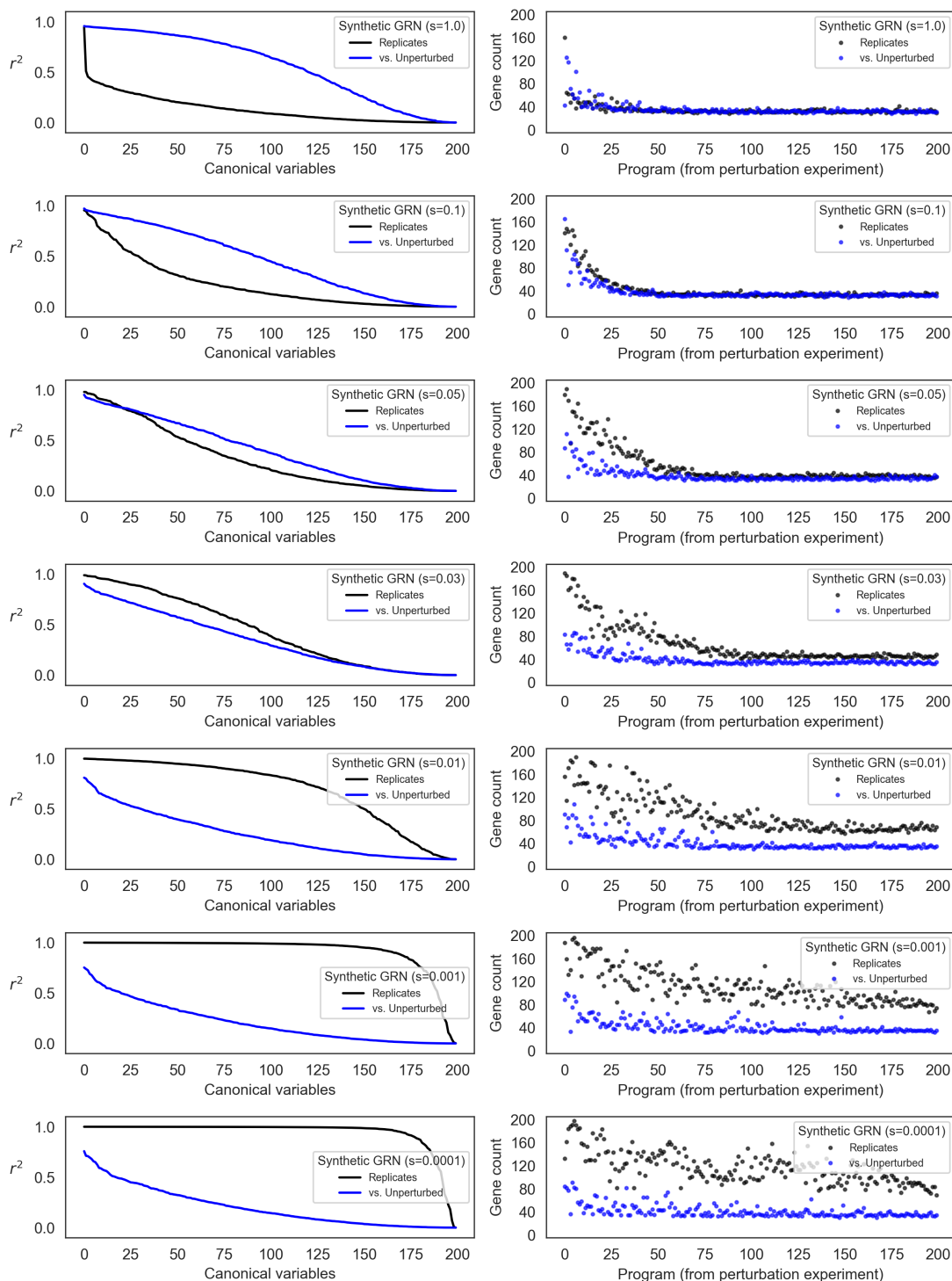
**Figure S12: Graph properties inform the extent to which perturbation effects more strongly enrich for edges than co-expression.** Performance of perturbation effects ( $x$ -axis,  $|\log_2FC|$ ) and co-expression ( $y$ -axis,  $r^2$  between genes) in identifying edge. As a summary performance measure, we compute the average precision (AP) score during binary classification of pairs of genes as being connected an edge (ignoring direction). All 1,920 networks in the study are shown in both panels, and colored by parameters of interest: the sparsity parameter ( $r$ , left) and the group affinity parameter ( $w$ , right). Across networks, sparsity and modularity degrade the performance of coexpression values, but enhance the performance of perturbation effects; but in every network, perturbation effects outperform coexpression (all points are below the dashed grey line,  $y = x$ ).



**Figure S13: True groups in the synthetic GRN are represented among gene programs.** In the focal GRN from Fig. 7, we show the overlap between each of the true groups ( $k = 10$ , shown as points in each of the bins on the  $y$ -axis) and its closest matching program (maximum overlap across all 50 gene sets, values shown on the  $x$ -axis). Points corresponding to the same true group are connected with a line spanning across  $y$ -axis bins. There is similar representation of all of the groups among the learned gene programs, regardless of input data type.



**Figure S14: Program replication depends on the number of cells.** Same as Fig 7C,D – instead of taking downsamples of unperturbed cells from Replogle *et. al.*, 2022, we here downsample the entire experiment to various study sizes. Here, the “entire experiment” is the normalized expression measurements of 5,247 genes in 932,593 control and intervened-upon cells which received one of the 5,247 perturbations in our analysis subset (**Methods**). We compare singular vectors (left) and programs (right) from the resulting downsamples of the entire experiment, as well as the subsets from Fig 7C,D. We note that the 75,328 control cells replicate the programs from the entire dataset comparably to 30,000 cells from the entire experiment.



**Figure S15: Program replication depends on the magnitude of intrinsic noise.** Same as Fig 7A,B for different levels of noise. We repeat CCA and analysis of gene programs as in Fig 7 (see **Methods**), varying the level of intrinsic noise ( $s$ ). At low levels of noise (small  $s$ ), replicates from perturbed conditions are much more similar to one another than to the unperturbed data. At high levels of noise (large  $s$ ), the perturbed data are more similar by canonical correlation to the unperturbed data than to the replicate perturbed data; but programs derived from each of the singular vectors are equivalently reproducible across conditions.