Gene regulatory network structure informs the distribution of perturbation effects

Matthew Aguirre[1], Jeffrey P. Spence[2], Guy Sella[3,4], Jonathan K. Pritchard[2,5]

[1] Department of Biomedical Data Science, Stanford University, Stanford CA
[2] Department of Genetics, Stanford University, Stanford CA
[3] Department of Biological Sciences, Columbia University, New York NY
[4] Program for Mathematical Genomics, Columbia University, New York NY
[5] Department of Biology, Stanford University, Stanford CA

Correspondence to:
magu@stanford.edu, jspence@stanford.edu,
gs2747@columbia.edu, pritch@stanford.edu

July 4, 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.

1 Introduction

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

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

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

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

Here, we characterize in detail a set of structural properties that we consider to be highly relevant for the study of GRNs. We propose a new algorithm to generate synthetic networks with these properties and formulate a gene expression model to simulate data from them. We use this simulation framework to conduct an array of *in silico* functional genomic studies and characterize the parameter space of our model in light of a recent genome-wide Perturb-seq study [9]. Our results provide intuition about the effects of various graph properties and how they manifest in experimental data. We conclude by discussing implications for future efforts to map

the architecture of gene regulation and complex traits, with particular emphasis on identifying pairwise regulatory relationships between genes and clustering genes into programs. Our analysis tools are available on github as a resource to the scientific community.

2 Main

2.1 Modeling approach

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

1. **GRNs are sparse**: While gene expression is controlled by many variables, the typical gene is directly affected by a small number of regulators. We further expect the number of regulators of any single gene to be much smaller than the total number of regulators in the network. Also, not all genes participate in expression regulation: only 41% of perturbations that target a primary transcript have significant effects on the expression of any other gene (**Fig. 1A**).

2. **GRNs have directed edges and feedback loops**: Regulatory relationships between genes are directed, with one gene acting as a regulator and the other as a target gene: this means that "A regulates B" is distinct from "B regulates A". Meanwhile, feedback loops are also thought to be pervasive in gene regulatory networks. A simple case of a feedback loop is bidirectional regulation, which is observed in data: 3.1% of ordered gene pairs have at least a one-directional perturbation effect (i.e., "A affects B", Anderson-Darling FDR-corrected $p < 0.05$), and 2.4% of these pairs further have bi-directional effects (i.e., "B also affects A") (**Fig. 1B**).

3. **GRNs have asymmetric distributions of in- and out-degree**: A further asymmetry between regulators and target genes arises from the existence of master regulators, which directly participate in the regulation of many other genes. The number of regulators per gene and genes per regulator are both thought to follow an approximate power-law distribution [20, 21], and indeed, the number of perturbation effects per regulator has a heavier-tailed distribution than the number of effects per target gene (**Fig. 1C**).

4. **GRNs are modular**: Genes in regulatory networks have different molecular functions that are executed in concert across various cell and tissue types. This grouping of genes by function also corresponds to a hierarchical organization of regulatory relationships that is revealed when these programs respond similarly to certain sets of perturbations (**Fig. 1D**).

While these criteria are not exhaustive, they do substantially constrain the space of plausible GRN structures. But from first principles, it is not obvious how these properties of GRNs manifest

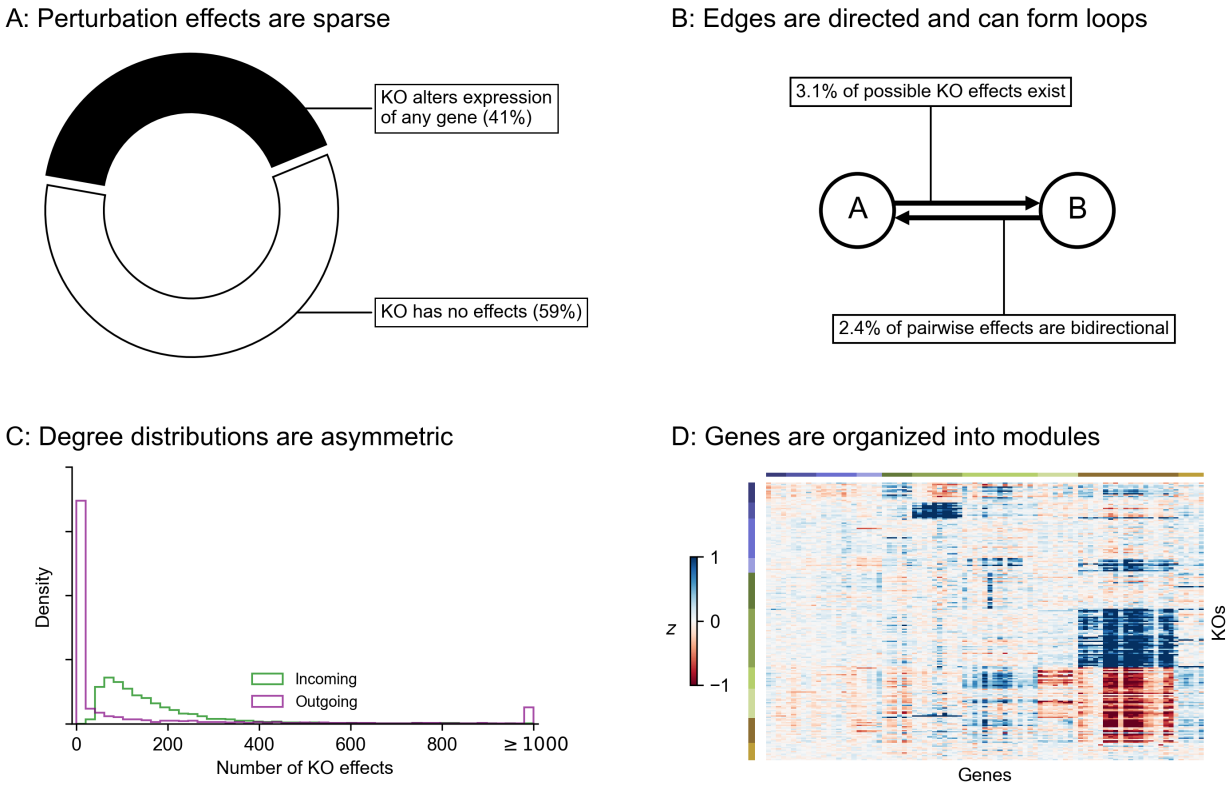

Figure 1: **Key properties of gene regulatory networks**. Data from Replogle *et. al.*, 2022. **(A)** Of the 5,247 perturbations in our analysis subset, 2,152 (41%) have a measurable effect on the transcriptional state of cells (energy-test $p < 0.001$). **(B)** Among all ordered pairs of genes, 3.1% (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 ± 1, for visualization.

in an ever-growing body of experimental data. In other words, what does it matter that networks are sparse, or modular?

2.2 Network generating algorithm

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

Our algorithm is based on that of Bollobas *et. al.*, 2003 [28], which models network growth

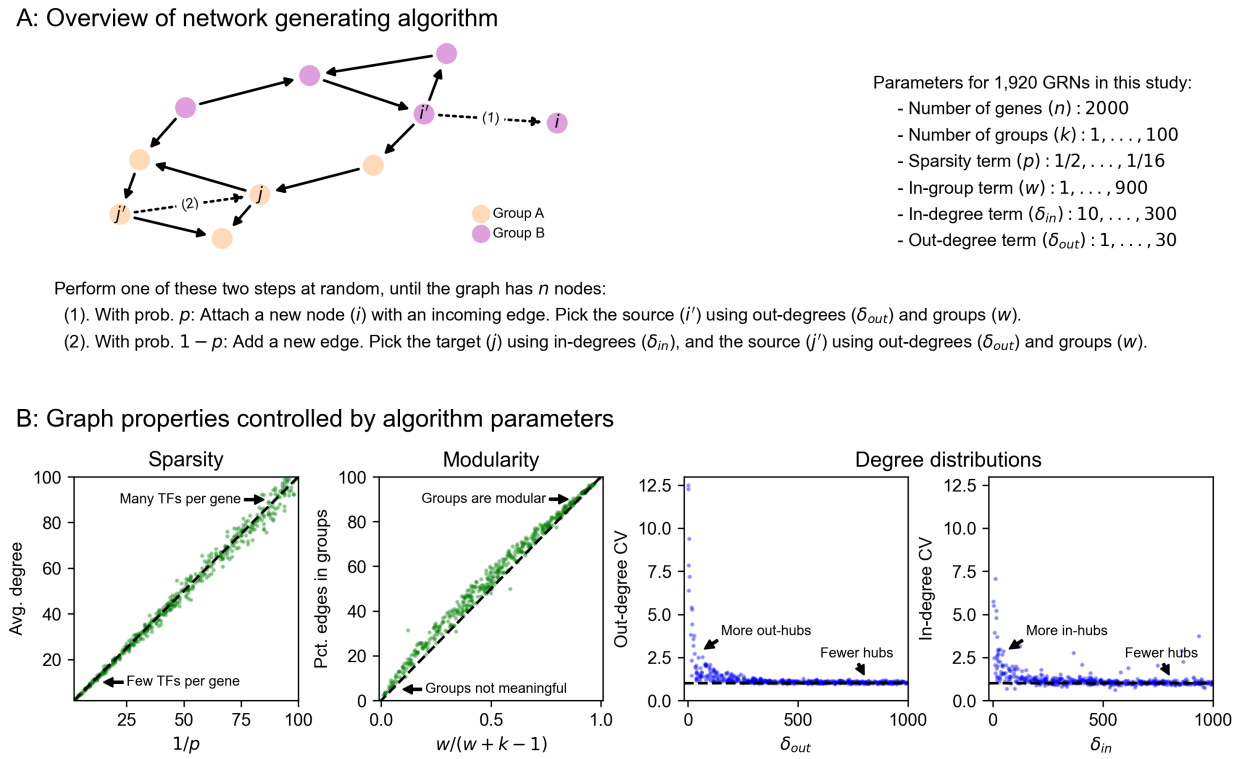

Figure 2: **Modeling approach and network generating algorithm**. **(A)** Overview of network generating algorithm, based on a growth process with preferential attachment. At each step, randomly add either a node or an edge, with the source and target determined by the out- and in-degree distributions, 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.

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

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

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 ($0 < p \leq 1$).
- δ_{in}: In-degree biasing term ($\delta_{in} \geq 0$).
- δ_{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.* ◁

$G \leftarrow \{(1 \to 2), (2 \to 3), (3 \to 1)\}$

$\text{gp}(i) \leftarrow i, \quad i \in \{1, 2, 3\}$ ▷ *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 `runif(0,1) < p` **then**

    $i \leftarrow |G| + 1$

    $\text{gp}(i) \leftarrow g \in \{1, \ldots, k\}$ uniformly at random.

  else

    $i \leftarrow i \in \{1, \ldots, |G|\}$ 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.* ◁

  $j \leftarrow j \in \{1, \ldots, |G|\}$ 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 $(j \to i)$ to the graph. Note that j and i may be the same node, in which case we add the edge $(j \to j)$; j and i may also already share the edge $(j \to i)$, in which case we add a duplicate edge.* ◁

  $G \leftarrow (j \to i)$

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

2.3 Expression model

In order to enable reasonable comparisons with experimental data, we use an expression model with quantitative (rather than binary) measurements, and with dynamics subject to a non-linearity that enforces realistic physical constraints: gene expression is non-negative and saturates near a

maximum value. Given a graph structure generated using the algorithm above, we assign parameters to each gene (node) and regulatory interaction (edge) in the graph. Each gene i has two rate parameters: one for innate RNA production in the absence of regulators (α_i), and another for the decay of existing cellular RNAs (ℓ_i). Each regulatory relationship, between genes j and i, has one parameter: a magnitude (β_{ji}) which describes the importance of the regulator for the expression of the target gene. We also enforce a constraint that interactions have a minimum strength ($|\beta_{ji}| \geq 1$). A full description of the strategy we use to sample these parameters for synthetic GRNs is in **Methods**.

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

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

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

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

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

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

2.4 Perturbation studies

We conducted synthetic perturbation studies in 1,920 GRNs with $n = 2{,}000$ genes – these GRNs were produced with a range of network generating parameters (**Methods**). For each GRN, we initialized gene expression values at zero and conducted a minimum 5,000 iterations of forward simulation, later verifying that the dynamical system reached equilibrium and assessing its stability (**Fig. 3A,B, Methods**). We then independently knocked out each gene in the network and let the system re-equilibrate after additional rounds of forward simulation (**Fig. 3B**). We computed

the effect of perturbing gene j as the log-fold change in expression x_i of all other genes i, which we refer to as the "perturbation effect" of gene j on gene i. Mathematically, this is

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

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

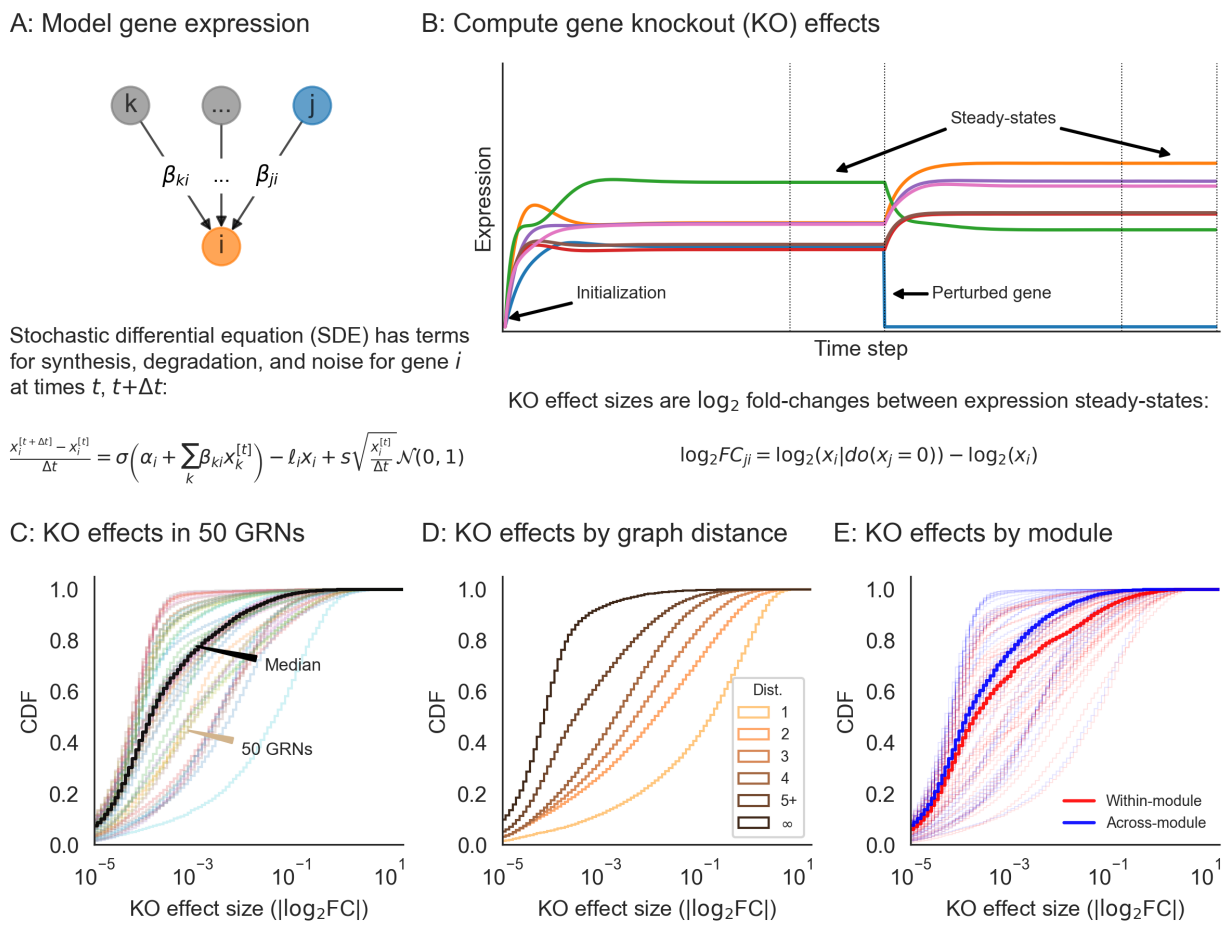

Figure 3: **Perturbations and their effects within networks**. **(A)** Overview of gene expression model and its parameters. Here, σ is the logistic sigmoid $\sigma(x) = 1/(1 + e^{-x})$. **(B)** Example forward simulation of the dynamical systems model. Trace lines show genes, whose expression values 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 ($\log_2 \mathrm{FC}$ 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.

These effects are largely stratified by the distance between regulator and target (**Fig. 3D**), with

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

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

2.5 Impact of network properties

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

Graph sparsity has the greatest influence on the number of hub KOs and hub target genes in the GRNs (**Fig. 4A**). More regulators per gene (large $1/p$) tends to translate to more perturbation effects overall, increasing the number of both hub knockouts and target genes. Notably, the effects on regulators and targets are not identical. In denser networks, the median number of hub KOs tends to be larger than the number of hub target genes. However, in a subset of dense networks, most genes in the network are identified as hub target genes. This is related to the absence of stable equilibrium dynamics in the low-noise limit of the gene expression model (**Fig. S4**), which suggests that as GRNs become more dense and genes are subject to regulation by larger fractions of the network, the system is less likely to be stable. Although this term has a large effect on perturbation effects, we find no obvious interaction between it and other terms in the generating algorithm (**Fig. S5**).

GRNs with fewer groups (small k) and higher modularity (large w) tend to have fewer hub KOs and hub target genes (**Figs. 4B, 4C**). The modularity term monotonically increases resilience

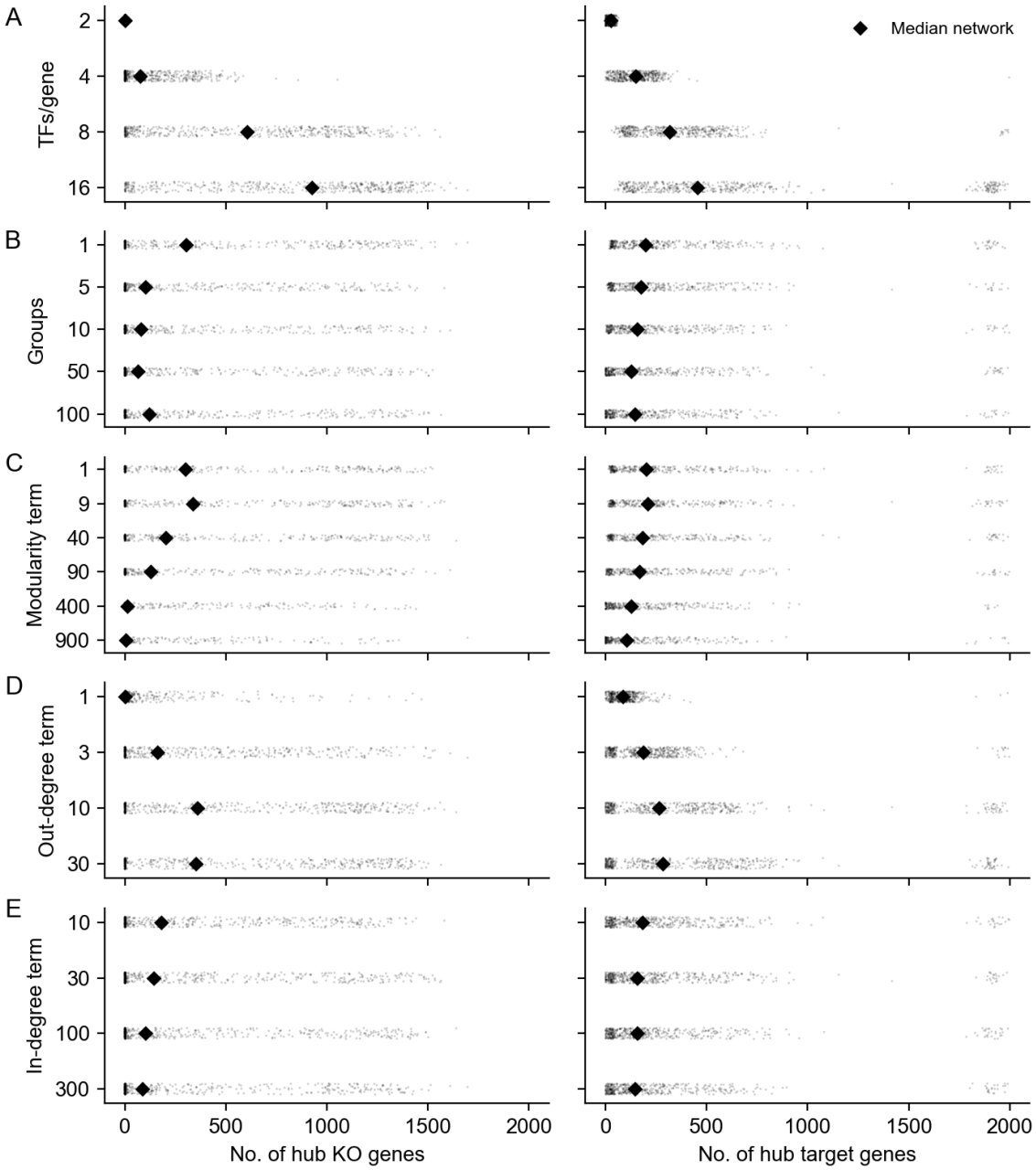

Figure 4: **Network properties influence the distribution of perturbation effects**. Counts of genes that are hub knockouts **(left)** and hub target genes **(right)** in each synthetic GRN, as a function of network generating parameters. Each panel shows all 1,920 GRNs as individual points, stratified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

to perturbation; the group term monotonically decreases it, with the exception of $k = 1$. From the perspective of the network generating algorithm, $k = 1$ and $w = 1$ are identical; they are equivalent to the algorithm from Bollobas *et. al.*, 2003 [28] and correspond to the dissolution of modular structure with respect to the specified grouping. This is also equivalent to $k = 2000$, in which each gene in the network has its own group – this intuition is supported by the remaining

trend in (**Fig. 4B**). Meanwhile, in modular networks (large w), most edges are between members of the same group. This might serve to confine the downstream effects of perturbations to members of the same group, effectively dampening the transcriptional impact of altering the function of master regulators.

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

Looking across parameters, these results reflect a wide range of variation in the susceptibility of GRNs to perturbations as a function of their structural properties. While there is substantial overlap in the distributions of hub KO and target genes across network generating parameters, we find that all parameters except the in-degree term have statistically significant effects on both quantities ($p < 0.001$ for all tests, **Fig. S7** – full results in **Tables S1, S2**). We estimate these effects with a multiple regression on the logit-transformed fraction of genes in each GRN which are hub KOs or hub target genes (**Methods**). In total, the network generating parameters explain just under half the variance in the fraction of the GRN which is either a hub KO (model $r^2 = 0.59$) or hub target gene (model $r^2 = 0.46$). Moreover, there is also a noteworthy thematic consistency across parameters. In all cases, the direction of protective effect from perturbation is also the direction of biological plausibility with respect to our modeling desiderata, where intuition dictates that GRNs should be sparse, modular, and have a heavy-tailed out- but not in-degree distribution.

2.6 Comparing with experimental data

With this intuition about network properties in hand, we now return to real experimental data. Given that synthetic GRNs with quantifiably different structures produce qualitatively different distributions of knockout effects, we next sought to ask whether any of them were also similar to real data. For this, we made use of the subset of perturbations that correspond one-to-one with gene expression readout in a recent Perturb-seq study [9].

We compared the real and synthetic data using their cumulative distributions of perturbation effects, computed both from the perspective of genes as regulators and as targets of regulation (**Fig. 5A-B**). Since these data have different numbers of genes that are not lowly expressed, we normalized the number of incoming and outgoing effects to the size of the network (**Methods**). In the Perturb-seq data, we find noticeable qualitative differences between the distribution of incoming and outgoing effects – these differences are consistent with the notion that GRNs should have master regulators, but not master target genes. In the synthetic data, we find substantial diversity in both distributions across GRNs, including many patterns that seem wholly incompatible with

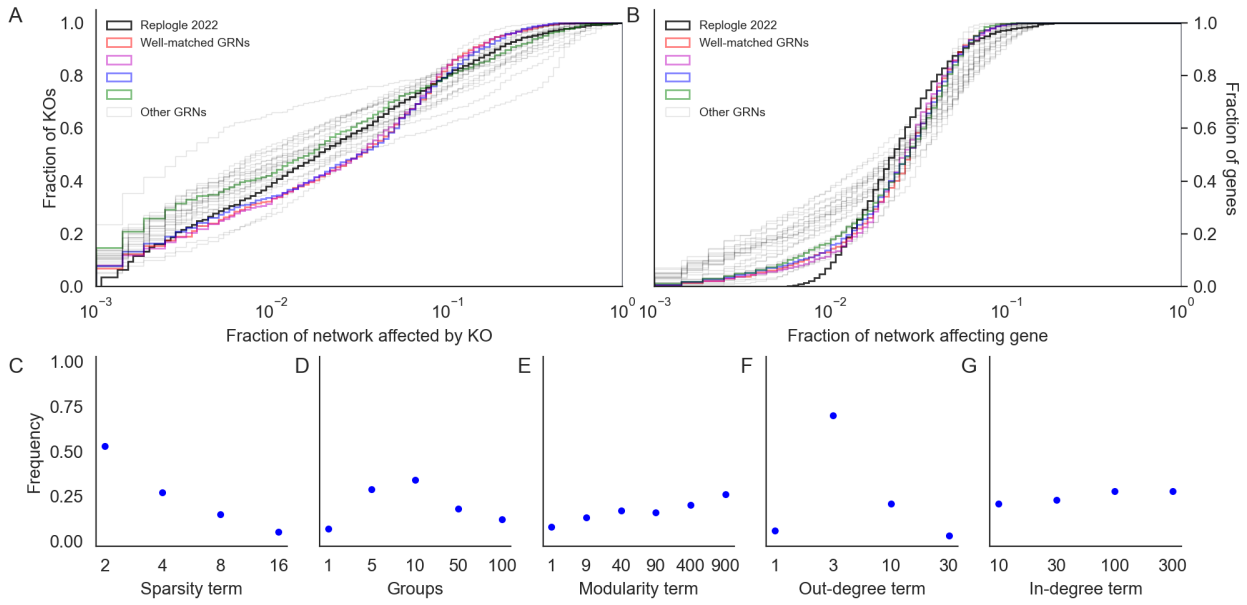

Figure 5: **Comparing with genome-wide Perturb-seq**. Fraction of GRN that **(A)** affects each gene when perturbed (outgoing effects) or **(B)** is affected by other perturbations (incoming effects). In synthetic data, perturbation effects are thresholded at the top 3% of absolute log-fold change values, 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).

those observed in experiments. Meanwhile, some GRNs seem well-matched to the Perturb-seq results: the distributions closest to the data are highlighted in color in **Fig. 5A-B**, and correspond to those having the smallest Kolmogorov-Smirnov test statistics when compared with the data distributions (**Methods**).

Although the focus of our work is not network inference, we do observe a coherent set of properties among the well-matched networks (**Fig. 5C-G**). Specifically, they share a relatively small number of regulators per gene (two, rather than 16); they have a small number of groups (five to ten rather than one or 100); they are highly modular (large w); and they have a heavy tail in the distribution of out-degree but not in-degree (δ_{out} near three but δ_{in} on the order of 100). Consistent with previous results, we find these parameter sets to be within a range that matches our motivating intuition about the structural properties of GRNs, and we do not find these properties to have obvious pairwise interactions that affect concordance to data (**Fig. S8**).

2.7 Challenges and opportunities for inference

Finally, we highlight the utility of our simulation approach by considering the value of different data sources for inference tasks. For this, we conducted further analysis using an example synthetic GRN whose patterns of knockout effects were well-matched with Perturb-seq data. Specifically, we focused on the recovery of edges, edge weights, and group structure using interventional data (e.g., perturbation effects) and observational data (e.g., coexpression). We made use

of perturbation effects as previously described, and further computed pairwise gene coexpression values using additional rounds of forward simulation from the expression model at steady state to approximate the naturally occurring variation across cells (**Methods**).

2.7.1 Discovering pairwise relationships

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

In the synthetic data, we find that the distributions of pairwise gene coexpression values and knockout effects both span multiple orders of magnitude (**Fig. 6**). However, where the distribution of knockout effects differs dramatically between gene pairs that share an edge and those that do not, the distributions of coexpression values have substantial overlap (**Fig. 6A,B**). This difference in distribution reflects what each statistic tends to measure. Gene perturbation effects tend to flow through the network along edges, and are therefore highly related to the network distance between genes (**Fig. 3D**). As a special case, this includes whether or not two genes share a direct regulatory relationship in the form of an edge. Meanwhile, strong coexpression is more often due to co-regulation than to there being a direct causal relationship between genes (**Fig. S9**).

For gene pairs where there is direct regulation we see that both knockout effects (**Fig. 6C**) and coexpression (**Fig. 6D**) have weak correlation with the strength of known regulatory relationships. This reflects that both statistics are imperfect measures of regulatory importance: they are both affected by differences in regulatory architecture across genes (e.g., number of regulators or the intensity of transcriptional buffering). We further see this when directly comparing coexpression and knockout effects between pairs of genes. Across all pairs of genes, these two statistics are uncorrelated – but the two are highly correlated among pairs of genes that share an edge (**Fig. 6E**). In this way, both perturbation effects and coexpression contain similar information about edges in the GRN – but coexpression also measures non-causal relationships between genes, like coregulation, and is therefore systematically uncorrelated with perturbation effects even in real data (**Fig. S10**).

Together, these results underscore the importance of interventional data for inferring network edges. They also highlight limitations in the use of coexpression networks. But neither form of data are a panacea, and care is warranted in the analysis of experimental data and in the development of structure learning algorithms. For example, sorting and thresholding perturbation effects has been shown to be a high-quality baseline for network reconstruction [10, 11] (one that we also use to compare networks in **Fig. 5**). However, false negative effects can arise when regulatory effects are weak due to transcriptional buffering, and false positive edges can be drawn where effects are amplified by mediation. This suggests that structure learning algorithms could benefit from modeling the transcriptional state of individual genes (i.e., level of buffering at baseline) and from approaches to explicitly resolve direct versus mediated effects.

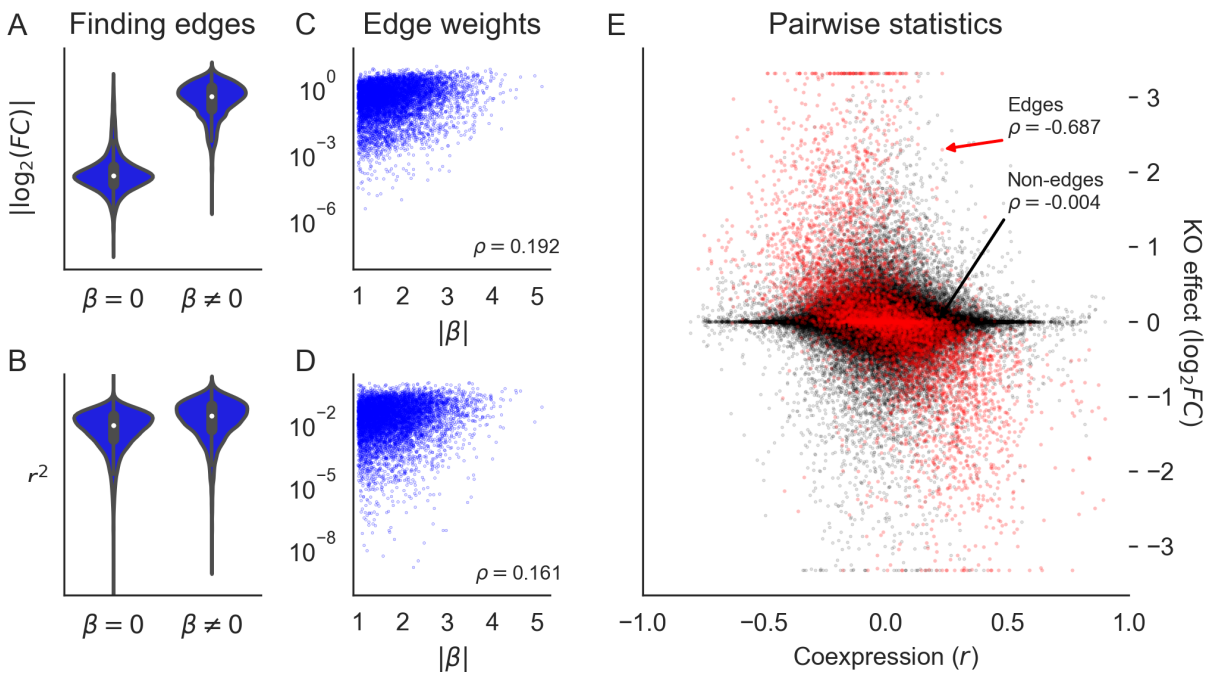

Figure 6: **Perturbations more reliably measure fine-scale network structure than coexpression**. **(A)** Distribution of perturbation effects between pairs of genes in a realistic synthetic GRN that do or do not share an edge. **(B)** Rank correlation of perturbation effect sizes with edge weights. **(C)** Distribution of gene coexpression between pairs of genes that do or do not share an edge. **(D)** Rank correlation of coexpression magnitude with edge weights. **(E)** Rank correlation between coexpression and perturbation effects (the y-axis is clipped at values corresponding to tenfold change).

2.7.2 Discovering group structure

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

Here, we assess the extent to which these programs and their constituent singular vectors replicate across experiments from perturbed and unperturbed conditions. We use canonical correlation analysis (CCA) to assess the similarity of the singular vectors. This technique seeks to find rotations a, b of inputs x, y such that the correlation between $a^\top x$ and $b^\top y$ is maximized – the transformed inputs are called canonical variables, and we report their correlations when the inputs x, y are gene singular vectors from different experimental conditions (**Fig. 7A**). In the syn-

thetic GRN, the canonical correlation steadily declines over the 200 dimensions of input. Notably, the magnitude of this correlation is similar when perturbation data are compared to a replicate or to unperturbed data. Even though this correlation is modest by the 100th set of canonical variables, this trend suggests consistency between lower-dimensional representations of expression data regardless of cell state (perturbed or unperturbed).

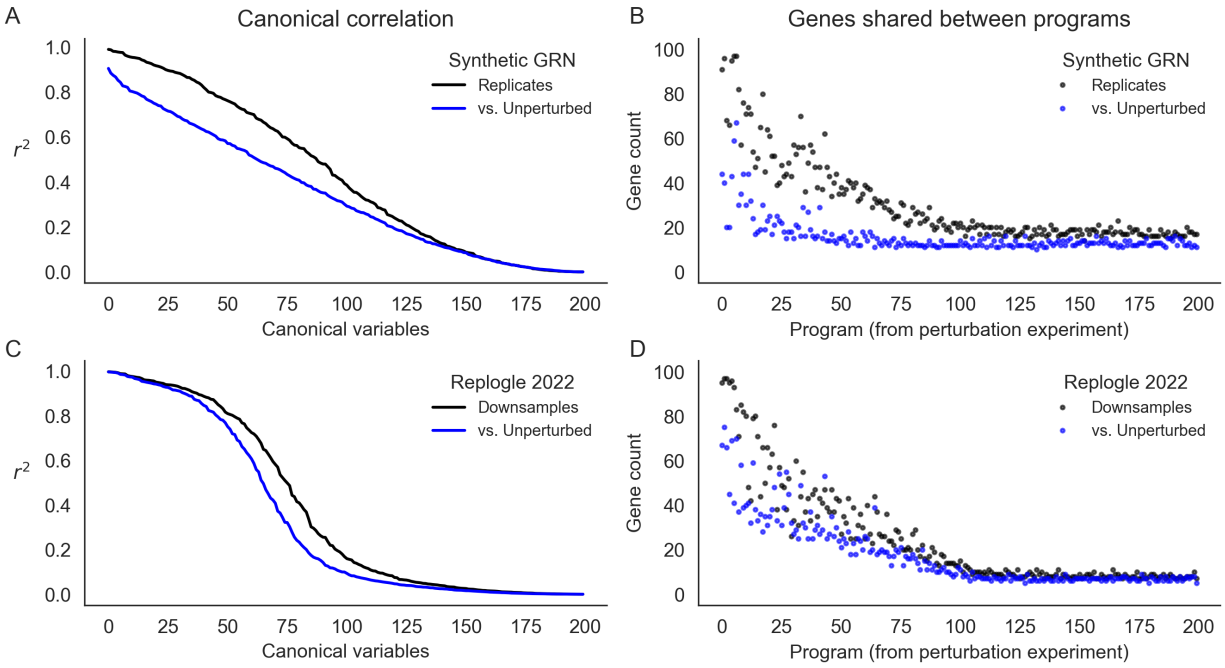

Figure 7: **Learned representations are similar between control and intervened-upon cells**. Concordance between low-rank representations of single cells in a simulated GRN (top row) and downsamples of experimental data (Replogle *et. al.*, 2022). **(A)** Correlation between the first 200 canonical variables (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.

At this sample size (75,238 cells) for the synthetic GRN, however, there is a difference in the reproducibility of individual programs across data sources (perturbed or unperturbed; **Fig. 7B**). For this, we compare programs computed from one perturbation experiment (the "reference") to programs from a replicate perturbation experiment and to programs from unperturbed cells. For each program from the reference, we report the maximum number of genes which overlap any other program computed from each of the other data sources. The first few programs (corresponding to the first few singular vectors) are highly reproducible in the replicate perturbation data. This overlap steadily declines to no effective overlap after the first $\sim$100 programs. Despite similar canonical correlation, the unperturbed data do not replicate the same programs to the same extent – there is modest overlap with the first few programs from the perturbation data, and this overlap decays very quickly (after the first $\sim$20 programs). When assessing these programs with respect

to the ($k = 10$) ground truth groups of this network, we find that nearly all true groups are at least modestly well represented by the top 50 programs from both data sources. However, the programs from the perturbation data much better represent the true groups than those from the unperturbed data (**Fig. S11**).

We find similar results in the experimental Perturb-seq data. Here, however, the first few canonical variables are highly correlated, and the canonical correlation drops off precipitously between the 50th and 100th canonical variables (**Fig. 7C**). We also find that programs computed from two downsamples of the entire experiment are about as reproducible as those from the simulated data, but are slightly more similar to the programs from control cells (**Fig. 7D**). While this may reflect some aspect of GRN structure, it is also related to the number of cells in the input data and magnitude of intrinsic gene expression noise. Both tend to reduce canonical correlations and the reproducibility of gene programs. In the real data, lowering the number of cells input to TSVD lowers both canonical correlation and program similarity compared to the entire experiment; relatedly, we find that the 75,328 control cells exhibit comparable performance as 30,000 perturbed cells at recovering representations from the full data (**Fig. S12**). In the synthetic networks, we find that altering the level of global transcriptional noise (s) alters the concordance between replicates and between perturbed and unperturbed cell states – at high levels of noise, there is little practical difference between programs derived from perturbed or unperturbed cells, but with low levels of noise, the programs from perturbation data are markedly different (**Fig. S13**). For presentation in **Fig. 7A-B**, we chose a level of noise ($s = 0.3$) that recapitulated the qualitative behavior of the real data.

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

3 Discussion

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

Here, we have highlighted the utility of our approach with simulations to develop intuition about key properties of GRNs, particularly in the context of molecular perturbations and coexpres-

sion data. While our study design draws inspiration from recent works using Perturb-seq, we also acknowledge limitations to the realism of our model. In focusing on the equilibrium dynamics of cells of one type, we have ignored developmental trajectories and cell-type heterogeneity within tissues, both of which modify our assumptions about regulatory network structure. For the sake of computational efficiency in quantifying expression for thousands of genes, we have also eschewed detailed models of the biological synthesis and experimental measurement of cellular RNAs: in future work where it is critical to match distributions of count data from experiments, we encourage modeling these complexities. Similar considerations may also be necessary for the application of our approach beyond modeling knockouts – for example, in studying genetic variation which affects gene expression.

Independently, our results suggest that the space of realistic network structures may be quite limited, and that it may be useful to consider this prior information in various inference settings. While our approach as outlined in this work is not optimized for inference, the algorithms we describe are generative, which means they could be used directly in applications for simulation-based inference. Although we used experimental data from K562 cells in this study, we anticipate the high-level structural properties of GRNs will generalize across different cell types. Moreover, we observed through simulations that hallmark properties of GRNs tend to confer resilience to perturbations across multiple measures, reducing the number of sensitive target genes and large-effect master regulators. We do not suspect this is an incidental finding in light of the selection to which GRNs are subject over evolutionary time, and suspect that considering this type of constraint may be insightful for future work.

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

4 Acknowledgements

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

5 Methods

5.1 Graph generating parameters

5.1.1 Sampling

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

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

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

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

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

5.1.2 Relationship to perturbation effects

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

following equation:

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

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

5.2 Expression simulation

5.2.1 Parameter selection

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

- the baseline transcription rate, α_i,
- the degradation rate of RNAs, ℓ_i,
- effects from regulating genes, β_{ji},
- expression noise, with magnitude s.

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

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

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

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

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

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

- $\beta_{ji} \overset{\text{iid}}{\sim} (2p_j - 1) \cdot (1 + \text{Half-Normal}(0, 1))$, under the assumptions that regulatory interactions have a minimum strength ($|\beta_{ji}| \geq 1$). Here, $p_j \overset{\text{iid}}{\sim} \text{Bernoulli}(p = 0.8)$ is the probability that a regulator j acts as an activator rather than a repressor.

- $s = 10^{-4}$, fixed across all genes in the networks. This value is chosen to be as large as possible without limiting detection of very small KO effects. At this level of noise, we can reliably detect $\log_2$ fold-changes down to the order of 10^{-4} (**Fig. 3D**).

5.2.2 Forward simulation

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

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

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

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

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

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

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

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

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

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

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

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

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

5.2.3 Perturbation experiments

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

We summarize the effect that perturbing gene j has on gene i using a log fold-change in expression values:

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

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

5.2.4 Baseline coexpression

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

5.3 Perturb-seq data

5.3.1 Data processing

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

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

5.3.2 Comparing with simulations

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

We then compared these distributions to analogous quantities derived from the synthetic GRNs. Since the experimental data are affected by imperfect statistical power, we set the discovery rate to be equal across all synthetic GRNs, doing so by taking the top 3.16% of pairwise perturbation effects (i.e., $|\log_2 \mathrm{FC}|_{ji} > k$, where k varies) as "statistically significant". For each gene i in a synthetic GRN, we computed the fraction of the network which is affected when i is perturbed (i.e., the fraction of genes j for which $|\log_2 \mathrm{FC}|_{ij} > k$), and the fraction of the network which affects i when perturbed (i.e., the fraction of genes j for which $|\log_2 \mathrm{FC}|_{ji} > k$). Note that in each GRN in **Fig. 5**, we remove lowly-expressed genes, with baseline expression $x_i < s$. This means that the number of genes analyzed is not exactly the same for all GRNs – we therefore normalized the distribution of perturbation effects by the number of genes that are included in the analysis (i.e., those not lowly-expressed).

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

5.4 Gene programs

5.4.1 Truncated singular value decomposition

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

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

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

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

$$X \approx USV^\top$$

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

5.4.2 Similarity across datasets

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

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

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

References

[1] Efthymia Papalexi and Rahul Satija. Single-cell RNA sequencing to explore immune cell heterogeneity. *Nature Reviews Immunology*, 18(1):35–45, January 2018. Publisher: Nature Publishing Group.

[2] Gioele La Manno, Ruslan Soldatov, Amit Zeisel, Emelie Braun, Hannah Hochgerner, Viktor Petukhov, Katja Lidschreiber, Maria E. Kastriti, Peter Lönnerberg, Alessandro Furlan, Jean Fan, Lars E. Borm, Zehua Liu, David van Bruggen, Jimin Guo, Xiaoling He, Roger Barker, Erik Sundström, Gonçalo Castelo-Branco, Patrick Cramer, Igor Adameyko, Sten Linnarsson, and Peter V. Kharchenko. RNA velocity of single cells. *Nature*, 560(7719):494–498, August 2018. Publisher: Nature Publishing Group.

[3] Volker Bergen, Marius Lange, Stefan Peidli, F. Alexander Wolf, and Fabian J. Theis. Generalizing RNA velocity to transient cell states through dynamical modeling. *Nature Biotechnology*, 38(12):1408–1414, December 2020. Publisher: Nature Publishing Group.

[4] Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P. Fulco, Livnat Jerby-Arnon, Nemanja D. Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychowdhury, Britt Adamson, Thomas M. Norman, Eric S. Lander, Jonathan S. Weissman, Nir Friedman, and Aviv Regev. Perturb-Seq: Dissecting Molecular Circuits with Scalable Single-Cell RNA Profiling of Pooled Genetic Screens. *Cell*, 167(7):1853–1866.e17, December 2016.

[5] Joseph M. Replogle, Thomas M. Norman, Albert Xu, Jeffrey A. Hussmann, Jin Chen, J. Zachery Cogan, Elliott J. Meer, Jessica M. Terry, Daniel P. Riordan, Niranjan Srinivas, Ian T. Fiddes, Joseph G. Arthur, Luigi J. Alvarado, Katherine A. Pfeiffer, Tarjei S. Mikkelsen, Jonathan S. Weissman, and Britt Adamson. Combinatorial single-cell CRISPR screens by direct guide RNA capture and targeted sequencing. *Nature Biotechnology*, 38(8):954–961, August 2020. Publisher: Nature Publishing Group.

[6] Jacob W. Freimer, Oren Shaked, Sahin Naqvi, Nasa Sinnott-Armstrong, Arwa Kathiria, Christian M. Garrido, Amy F. Chen, Jessica T. Cortez, William J. Greenleaf, Jonathan K. Pritchard, and Alexander Marson. Systematic discovery and perturbation of regulatory genes in human T cells reveals the architecture of immune networks. *Nature Genetics*, 54(8):1133–1144, August 2022. Number: 8 Publisher: Nature Publishing Group.

[7] Joshua S. Weinstock, Maya M. Arce, Jacob W. Freimer, Mineto Ota, Alexander Marson, Alexis Battle, and Jonathan K. Pritchard. Gene regulatory network inference from CRISPR perturbations in primary CD4+ T cells elucidates the genomic basis of immune disease. *bioRxiv*, page 2023.09.17.557749, October 2023.

[8] Gavin R. Schnitzler, Helen Kang, Shi Fang, Ramcharan S. Angom, Vivian S. Lee-Kim, X. Rosa Ma, Ronghao Zhou, Tony Zeng, Katherine Guo, Martin S. Taylor, Shamsudheen K. Vellarikkal, Aurelie E. Barry, Oscar Sias-Garcia, Alex Bloemendal, Glen Munson, Philine Guckelberger, Tung H. Nguyen, Drew T. Bergman, Stephen Hinshaw, Nathan Cheng, Brian Cleary, Krishna Aragam, Eric S. Lander, Hilary K. Finucane, Debabrata Mukhopadhyay, Rajat M. Gupta, and Jesse M. Engreitz. Convergence of coronary artery disease genes onto endothelial cell programs. *Nature*, 626(8000):799–807, February 2024. Number: 8000 Publisher: Nature Publishing Group.

[9] Joseph M. Replogle, Reuben A. Saunders, Angela N. Pogson, Jeffrey A. Hussmann, Alexander Lenail, Alina Guna, Lauren Mascibroda, Eric J. Wagner, Karen Adelman, Gila Lithwick-Yanai, Nika Iremadze, Florian Oberstrass, Doron Lipson, Jessica L. Bonnar, Marco Jost, Thomas M. Norman, and Jonathan S. Weissman. Mapping information-rich genotype-phenotype landscapes with genome-scale Perturb-seq. *Cell*, 185(14):2559–2575.e28, July 2022.

[10] Mathieu Chevalley, Jacob Sackett-Sanders, Yusuf Roohani, Pascal Notin, Artemy Bakulin, Dariusz Brzezinski, Kaiwen Deng, Yuanfang Guan, Justin Hong, Michael Ibrahim, Wojciech Kotlowski, Marcin Kowiel, Panagiotis Misiakos, Achille Nazaret, Markus Püschel, Chris Wendler, Arash Mehrjou, and Patrick Schwab. The CausalBench challenge: A machine learning contest for gene network inference from single-cell perturbation data, August 2023. arXiv:2308.15395 [cs, q-bio].

[11] Mathieu Chevalley, Yusuf Roohani, Arash Mehrjou, Jure Leskovec, and Patrick Schwab. CausalBench: A Large-scale Benchmark for Network Inference from Single-cell Perturbation Data, July 2023. arXiv:2210.17283 [cs].

[12] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using Bayesian networks to analyze expression data. In *Proceedings of the fourth annual international conference on Computational molecular biology*, RECOMB '00, pages 127–135, New York, NY, USA, April 2000. Association for Computing Machinery.

[13] David Heckerman. A Tutorial on Learning with Bayesian Networks. In Dawn E. Holmes and Lakhmi C. Jain, editors, *Innovations in Bayesian Networks: Theory and Applications*, pages 33–82. Springer, Berlin, Heidelberg, 2008.

[14] Yuhao Wang, Liam Solus, Karren Dai Yang, and Caroline Uhler. Permutation-based Causal Inference Algorithms with Interventions, November 2017. arXiv:1705.10220 [stat].

[15] Romain Lopez, Jan-Christian Hütter, Jonathan K. Pritchard, and Aviv Regev. Large-Scale Differentiable Causal Discovery of Factor Graphs, October 2022. arXiv:2206.07824 [cs, q-bio, stat].

[16] Douglas Yao, Loic Binan, Jon Bezney, Brooke Simonton, Jahanara Freedman, Chris J. Frangieh, Kushal Dey, Kathryn Geiger-Schuller, Basak Eraslan, Alexander Gusev, Aviv Regev, and Brian Cleary. Scalable genetic screening for regulatory circuits using compressed Perturb-seq. *Nature Biotechnology*, pages 1–14, October 2023. Publisher: Nature Publishing Group.

[17] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network Motifs: Simple Building Blocks of Complex Networks. *Science*, 298(5594):824–827, October 2002. Publisher: American Association for the Advancement of Science.

[18] Shai S. Shen-Orr, Ron Milo, Shmoolik Mangan, and Uri Alon. Network motifs in the transcriptional regulation network of Escherichia coli. *Nature Genetics*, 31(1):64–68, May 2002. Publisher: Nature Publishing Group.

[19] C. Seshadhri, Aneesh Sharma, Andrew Stolman, and Ashish Goel. The impossibility of low-rank representations for triangle-rich complex networks. *Proceedings of the National Academy of Sciences*, 117(11):5631–5637, March 2020. Publisher: Proceedings of the National Academy of Sciences.

[20] S. Balaji, Lakshminarayan M. Iyer, L. Aravind, and M. Madan Babu. Uncovering a Hidden Distributed Architecture Behind Scale-free Transcriptional Regulatory Networks. *Journal of Molecular Biology*, 360(1):204–212, June 2006.

[21] Mark B. Gerstein, Anshul Kundaje, Manoj Hariharan, Stephen G. Landt, Koon-Kiu Yan, Chao Cheng, Xinmeng Jasmine Mu, Ekta Khurana, Joel Rozowsky, Roger Alexander, Renqiang Min, Pedro Alves, Alexej Abyzov, Nick Addleman, Nitin Bhardwaj, Alan P. Boyle, Philip Cayting, Alexandra Charos, David Z. Chen, Yong Cheng, Declan Clarke, Catharine Eastman, Ghia Euskirchen, Seth Frietze, Yao Fu, Jason Gertz, Fabian Grubert, Arif Harmanci, Preti Jain, Maya Kasowski, Phil Lacroute, Jing Leng, Jin Lian, Hannah Monahan, Henriette O'Geen, Zhengqing Ouyang, E. Christopher Partridge, Dorrelyn Patacsil, Florencia Pauli, Debasish Raha, Lucia Ramirez, Timothy E. Reddy, Brian Reed, Minyi Shi, Teri Slifer, Jing Wang, Linfeng Wu, Xinqiong Yang, Kevin Y. Yip, Gili Zilberman-Schapira, Serafim Batzoglou, Arend Sidow, Peggy J. Farnham, Richard M. Myers, Sherman M. Weissman, and Michael Snyder. Architecture of the human regulatory network derived from ENCODE data. *Nature*, 489(7414):91–100, September 2012. Publisher: Nature Publishing Group.

[22] Arun J. Singh, Stephen A. Ramsey, Theresa M. Filtz, and Chrissa Kioussi. Differential gene regulatory networks in development and disease. *Cellular and Molecular Life Sciences*, 75(6):1013–1025, March 2018.

[23] Peter Langfelder and Steve Horvath. WGCNA: an R package for weighted correlation network analysis. *BMC Bioinformatics*, 9(1):559, December 2008.

[24] Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, June 1983.

[25] Thorben Funke and Till Becker. Stochastic block models: A comparison of variants and inference methods. *PLOS ONE*, 14(4):e0215296, April 2019. Publisher: Public Library of Science.

[26] Albert-László Barabási and Réka Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, October 1999. Publisher: American Association for the Advancement of Science.

[27] Petter Holme and Beom Jun Kim. Growing scale-free networks with tunable clustering. *Physical Review E*, 65(2):026107, January 2002. Publisher: American Physical Society.

[28] Béla Bollobás, Christian Borgs, Jennifer T Chayes, and Oliver Riordan. Directed scale-free graphs. In *SODA*, volume 3, pages 132–139, 2003.

[29] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998. Publisher: Nature Publishing Group.

[30] Reuven Cohen and Shlomo Havlin. Scale-Free Networks Are Ultrasmall. *Physical Review Letters*, 90(5):058701, February 2003. Publisher: American Physical Society.

[31] Payam Dibaeinia and Saurabh Sinha. SERGIO: A Single-Cell Expression Simulator Guided by Gene Regulatory Networks. *Cell Systems*, 11(3):252–271.e11, September 2020.

[32] Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. DAGs with NO TEARS: Continuous Optimization for Structure Learning, November 2018. arXiv:1803.01422 [cs, stat].

[33] Dylan Kotliar, Adrian Veres, M Aurel Nagy, Shervin Tabrizi, Eran Hodis, Douglas A Melton, and Pardis C Sabeti. Identifying gene expression programs of cell-type identity and cellular activity with single-cell RNA-Seq. *eLife*, 8:e43803, July 2019. Publisher: eLife Sciences Publications, Ltd.

[34] Aviv Regev, Sarah A Teichmann, Eric S Lander, Ido Amit, Christophe Benoist, Ewan Birney, Bernd Bodenmiller, Peter Campbell, Piero Carninci, Menna Clatworthy, Hans Clevers, Bart Deplancke, Ian Dunham, James Eberwine, Roland Eils, Wolfgang Enard, Andrew Farmer, Lars Fugger, Berthold Göttgens, Nir Hacohen, Muzlifah Haniffa, Martin Hemberg, Seung Kim, Paul Klenerman, Arnold Kriegstein, Ed Lein, Sten Linnarsson, Emma Lundberg, Joakim Lundeberg, Partha Majumder, John C Marioni, Miriam Merad, Musa Mhlanga, Martijn Nawijn, Mihai Netea, Garry Nolan, Dana Pe'er, Anthony Phillipakis, Chris P Ponting, Stephen Quake, Wolf Reik, Orit Rozenblatt-Rosen, Joshua Sanes, Rahul Satija, Ton N Schumacher, Alex Shalek, Ehud Shapiro, Padmanee Sharma, Jay W Shin, Oliver Stegle, Michael Stratton, Michael J T Stubbington, Fabian J Theis, Matthias Uhlen, Alexander van Oudenaarden, Allon Wagner, Fiona Watt, Jonathan Weissman, Barbara Wold, Ramnik Xavier, Nir Yosef, and Human Cell Atlas Meeting Participants. The Human Cell Atlas. *eLife*, 6:e27041, December 2017. Publisher: eLife Sciences Publications, Ltd.

[35] THE TABULA SAPIENS CONSORTIUM. The Tabula Sapiens: A multiple-organ, single-cell transcriptomic atlas of humans. *Science*, 376(6594):eabl4896, May 2022. Publisher: American Association for the Advancement of Science.

[36] Colin Megill, Bruce Martin, Charlotte Weaver, Sidney Bell, Lia Prins, Seve Badajoz, Brian McCandless, Angela Oliveira Pisco, Marcus Kinsella, Fiona Griffin, Justin Kiggins, Genevieve Haliburton, Arathi Mani, Matthew Weiden, Madison Dunitz, Maximilian Lombardo, Timmy Huang, Trent Smith, Signe Chambers, Jeremy Freeman, Jonah Cool, and Ambrose Carr. cellxgene: a performant, scalable exploration platform for high dimensional sparse matrices, April 2021. Pages: 2021.04.05.438318 Section: New Results.

[37] Steven H Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. CRC press, 2018.

[38] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, March 2020. Publisher: Nature Publishing Group.

[39] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[40] William H Press. Canonical correlation clarified by singular value decomposition, 2011.

Supplementary Information

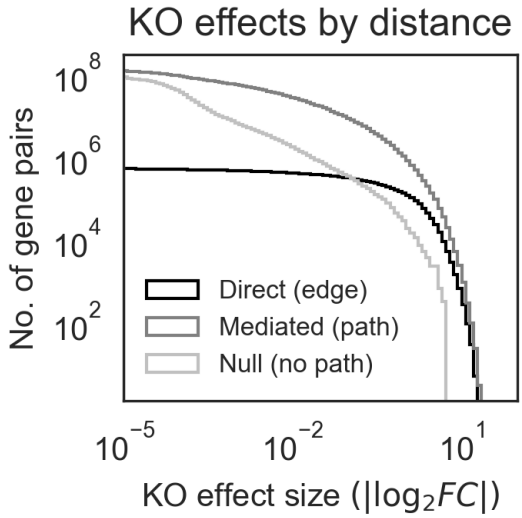

Figure S1: **Mediated effects outnumber direct effects at most magnitudes**. Same as Fig. 3D, but with distances binned by whether pairs of genes are connected by an edge (distance 1, a "direct effect"), any path (distance greater than 1, a "mediated effect"), or no path at all ("null"). Note also that 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.

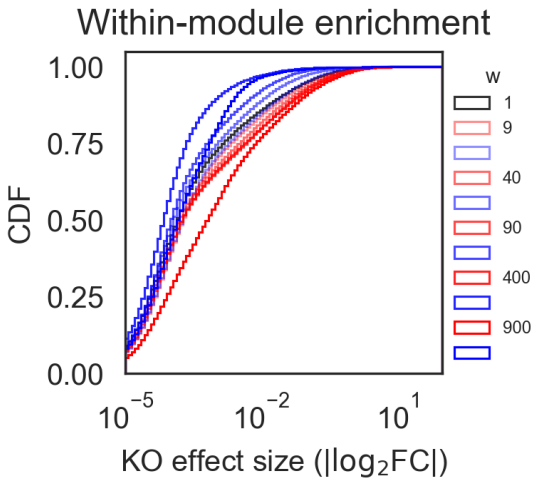

Figure S2: **Modularity term differentiates within- and between-module effects**. Same as Fig. 3E, with within-module perturbation effects in red and between-module perturbation effects in blue. Here, networks are chosen so as to highlight the effect of the modularity term w. Each pair of blue and red 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: $p = 1/4$, $k = 50$, $\delta_{in} = 10$, $\delta_{out} = 10$.

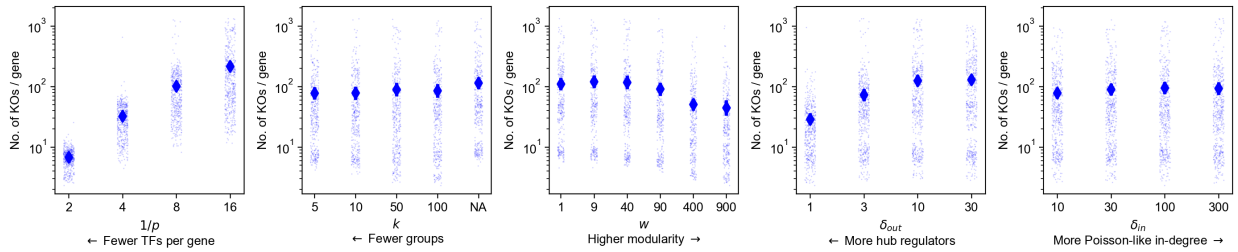

Figure S3: **Network generating parameters affect the number of KO effects**. Same as Fig. 4, but with summary statistic (y-axis) as the average number of perturbation effects per gene in the GRN with $|\log_2 \mathrm{FC}| \geq 0.1$. We observe a similar direction of effect for each parameter as with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

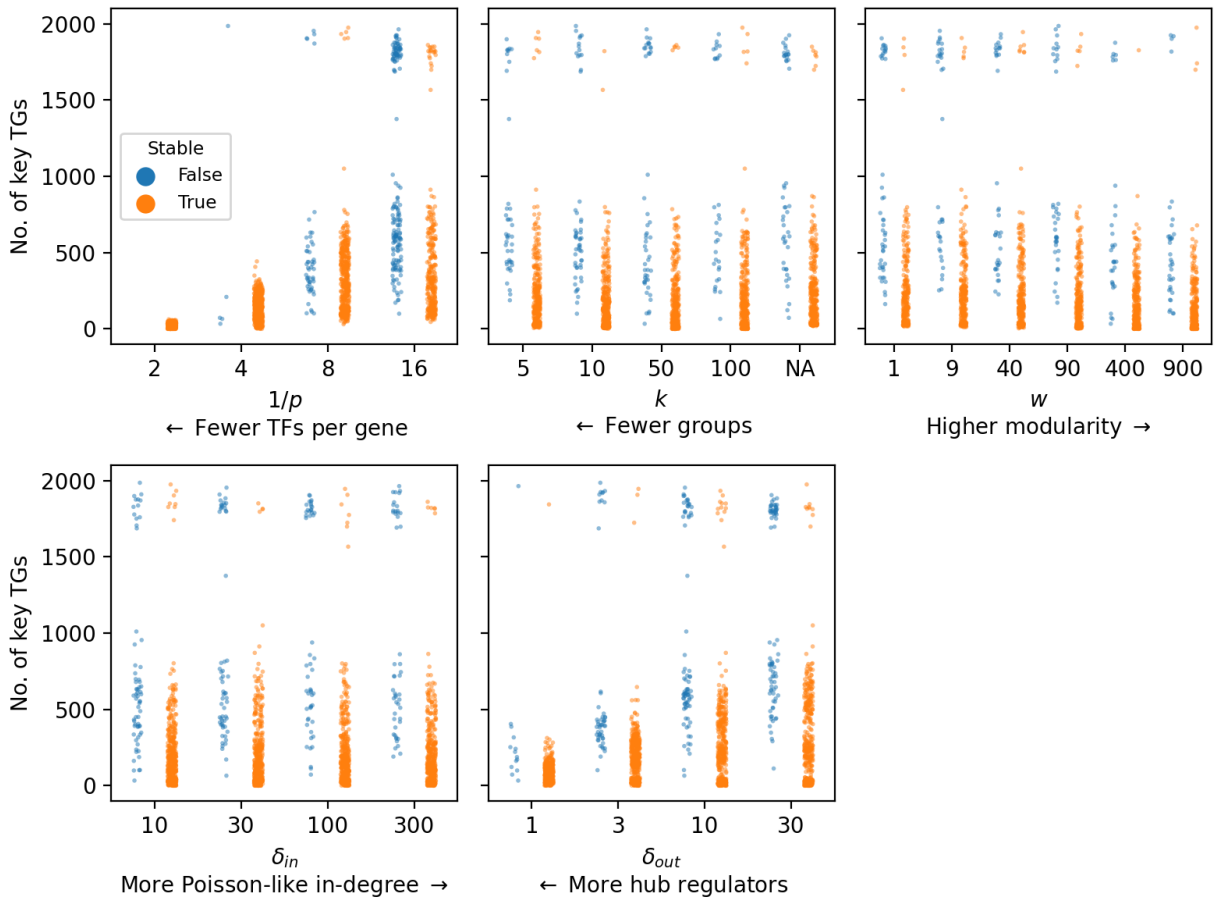

Figure S4: **Network generating parameters affect the stability of the fixed point**. Same as Fig. 4, only showing the number of key target genes and stratifying by whether the expression equilibrium point of the synthetic GRN is stable (**Methods**). In all, 1,693 of the 1,920 GRNs (88.2%) reach an 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 δ_{out}).

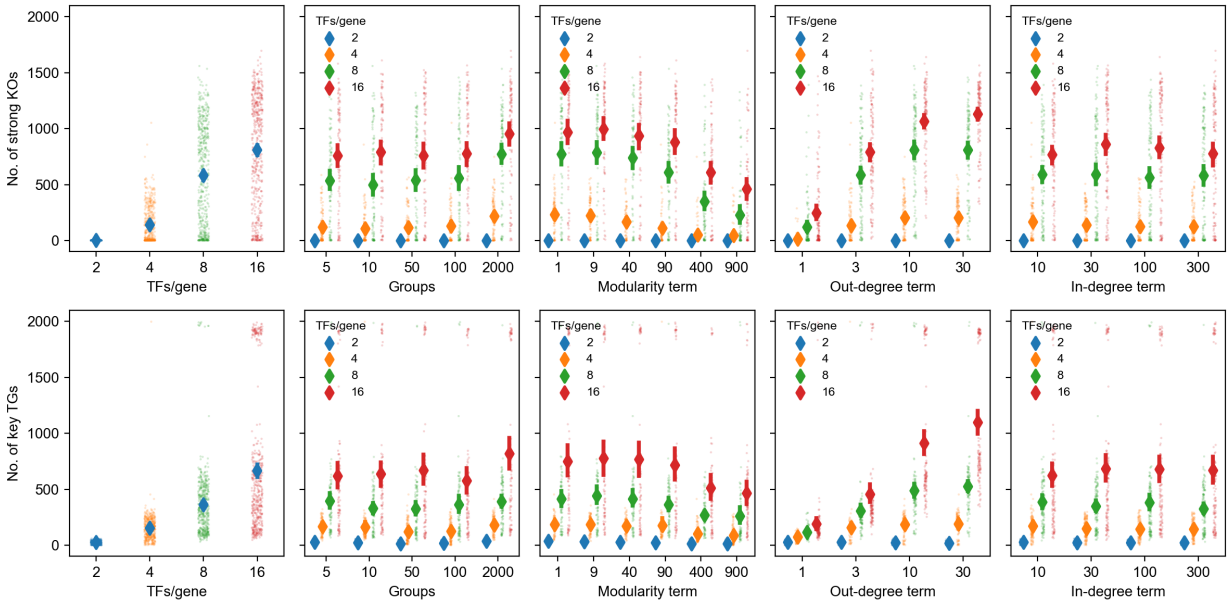

Figure S5: **No interaction between sparsity term and other network generating parameters**. Same as Fig. 4, but with additional stratification by the sparsity term $1/p$. There is no obvious visual evidence for interactions between the parameters.

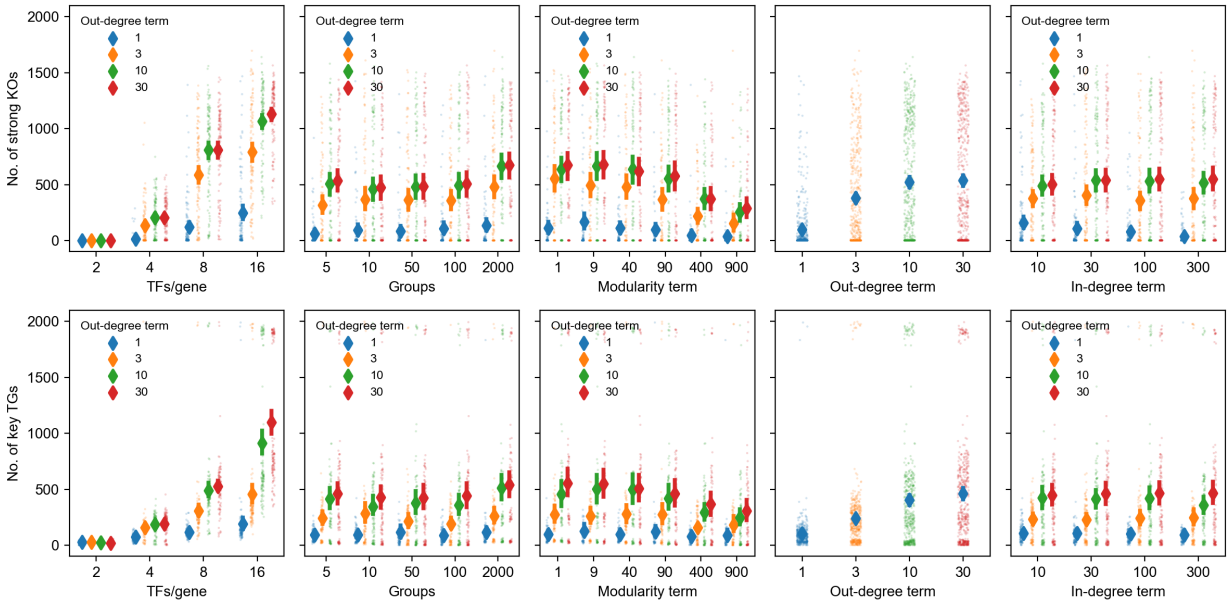

Figure S6: **No interaction between out-degree term and other network generating parameters**. Same as Fig. 4, but with additional stratification by the out-degree term δ_{out}. There is no obvious visual evidence for interactions between the parameters.

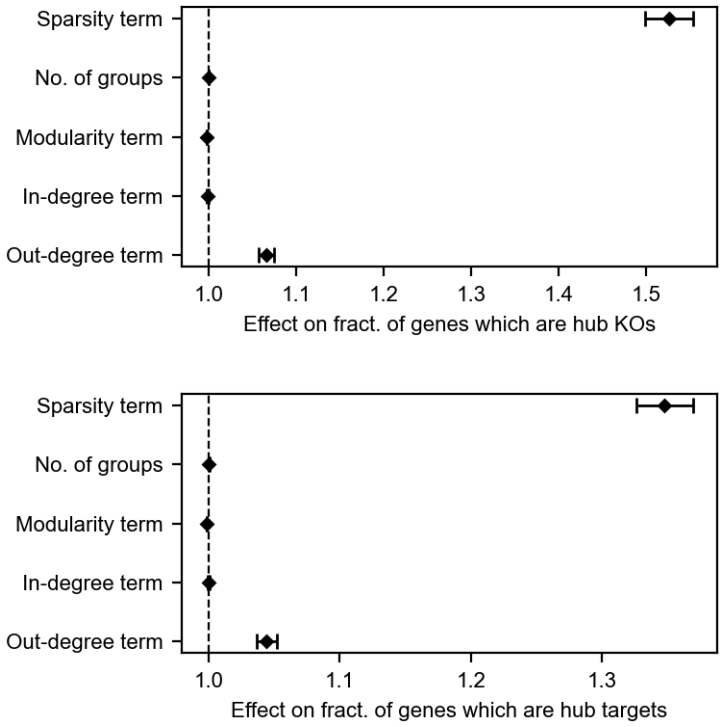

Figure S7: **Summary of regression models for effects of network parameters on perturbations**. Coefficients from regressing the logit-transformed fraction of genes which are hub knockouts (top) or target genes (bottom) on network generating parameters. Errorbars denote 95% confidence intervals for the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

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

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
const	-6.9629	0.115	-60.458	0.000	-7.189	-6.737
r	0.4229	0.009	46.789	0.000	0.405	0.441
k_adj	0.0004	6.18e-05	6.857	0.000	0.000	0.001
w	-0.0023	0.000	-15.724	0.000	-0.003	-0.002
delta_in	-0.0012	0.000	-2.801	0.005	-0.002	-0.000
delta_out	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).

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

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
const	-4.7854	0.104	-45.966	0.000	-4.990	-4.581
r	0.2983	0.008	36.514	0.000	0.282	0.314
k_adj	0.0003	5.59e-05	5.686	0.000	0.000	0.000
w	-0.0016	0.000	-12.002	0.000	-0.002	-0.001
delta_in	-0.0002	0.000	-0.438	0.661	-0.001	0.001
delta_out	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).

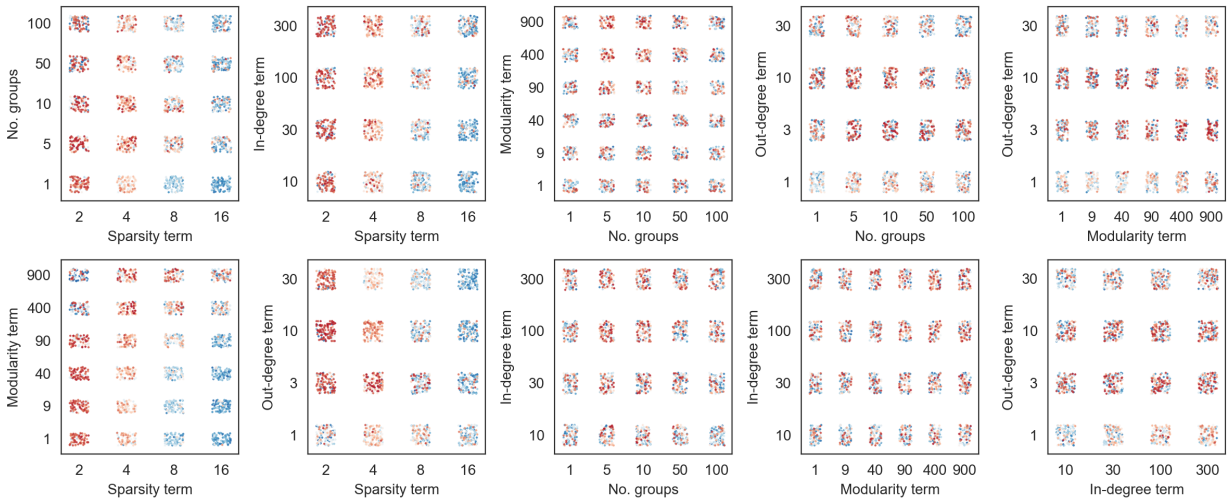

Figure S8: **No interaction between network generating parameters and fit to experimental data**. As in Fig. 5C-E, we show the relationship between pairs of network generating parameters and goodness of fit to the cumulative distribution of perturbation effects from experimental Perturb-seq data. Each 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.

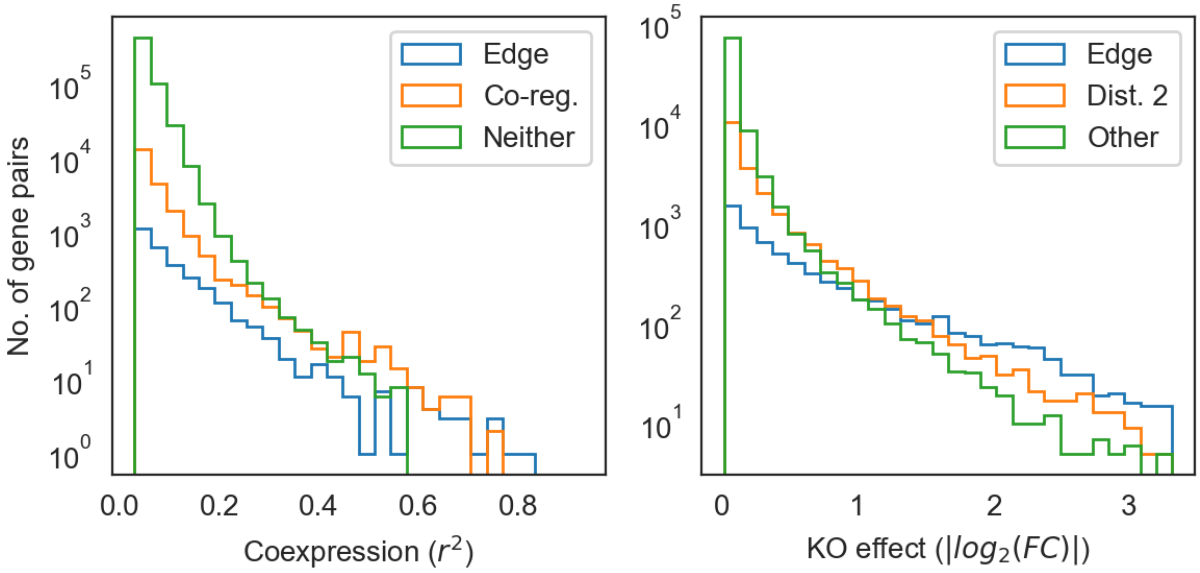

Figure S9: **Coexpression is more often due to coregulation than edges**. In the focal GRN from Fig. 6, we show a histogram of coexpression values split by whether pairs of genes share an edge ("A regulates B, or B regulates A", share a regulator ("A and B are coregulated"), or have another relationship (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**.

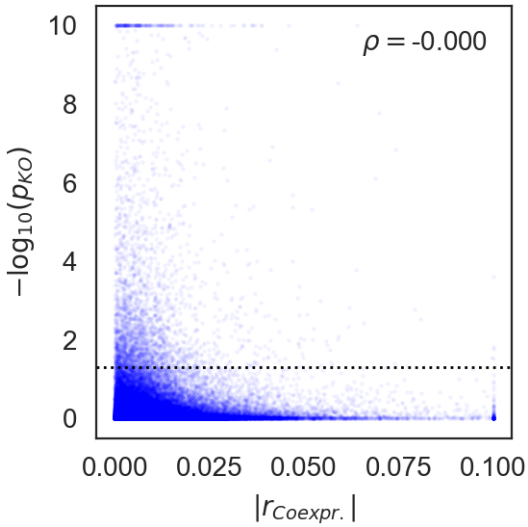

Figure S10: **Baseline coexpression and perturbation effects are uncorrelated in Perturb-seq data**. Same as **Fig. 6E**, using data from our analysis subset of Replogle *et. al.* 2022 [9]. Gene co-expression (x-axis) is the unsigned Pearson correlation between normalized single-cell gene expression 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 ρ) is computed on the transformed but not clipped values of these two statistics.

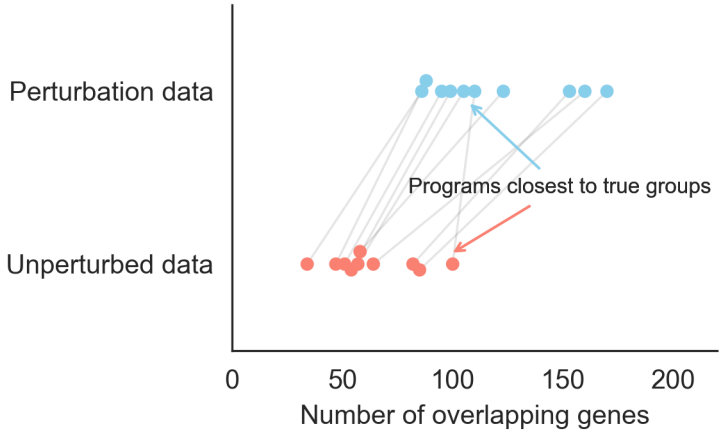

Figure S11: **True groups in the synthetic GRN are represented among gene programs**. In the focal GRN from Fig. 7, we show the overlap between each of the true groups ($k = 10$, shown as points in each of the bins on the y-axis) and its closest matching program (maximum overlap across all 50 gene sets, 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.

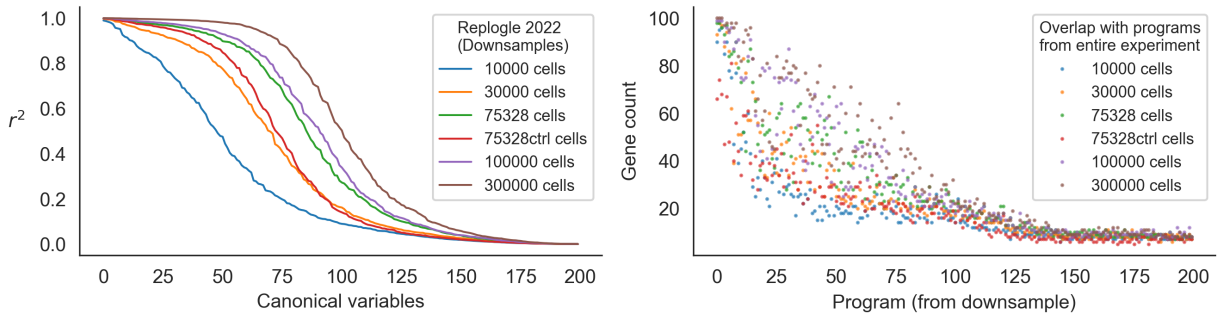

Figure S12: **Program replication depends on the number of cells**. Same as Fig 7C,D – instead of taking downsamples of unperturbed cells from Replogle *et. al.*, 2022, we here downsample the entire experiment to various study sizes. Here, the "entire experiment" is the normalized expression measurements 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.

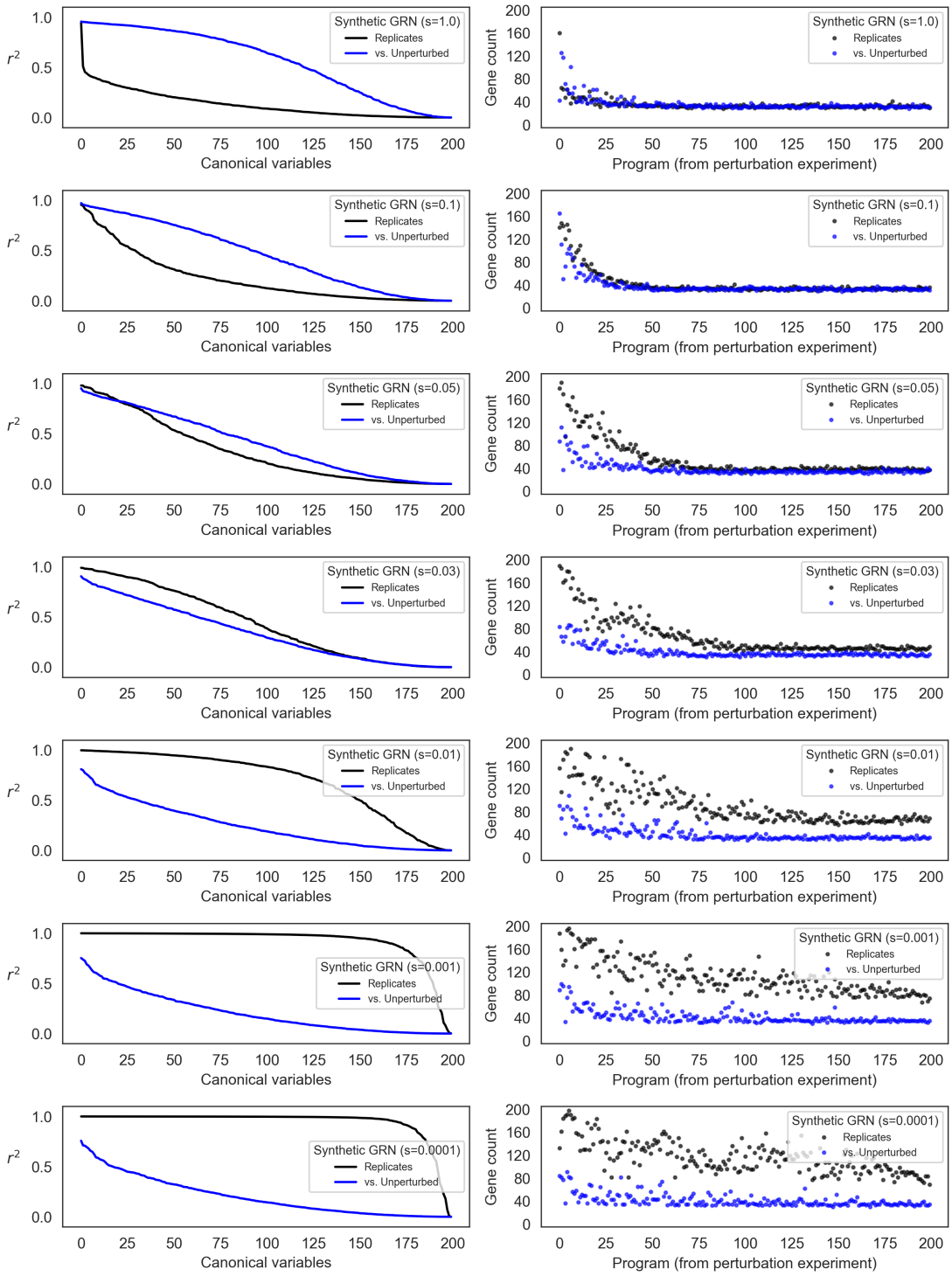

Figure S13: **Program replication depends on the magnitude of intrinsic noise**. Same as Fig 7A,B for different levels of noise. We repeat CCA and analysis of gene programs as in Fig 7 (see **Methods**), varying the level of intrinsic noise (s). At low levels of noise (small s), replicates from perturbed 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.