# D-SPIN constructs gene regulatory network models from multiplexed scRNA-seq data revealing organizing principles of cellular perturbation response

Jialong Jiang[1,*], Sisi Chen[1,2,11*], Tiffany Tsou[1,2], Christopher S. McGinnis[3], Tahmineh Khazaei[1], Qin Zhu[3], Jong H. Park[1,2], Inna-Marie Strazhnik[1], Jost Vielmetter[1], Yingying Gong[1], John Hanna[4], Eric D. Chow[5,6], David A. Sivak[7], Zev J. Gartner[3,8,9,10], and Matt Thomson[1,2†]

[1]Division of Biology and Biological Engineering, California Institute of Technology, Pasadena, California, 91125, USA
[2]Beckman Single-Cell Profiling and Engineering Center, California Institute of Technology, Pasadena, CA, 91125, USA
[3]Department of Pharmaceutical Chemistry, University of California San Francisco, San Francisco, CA, 94143, USA
[4]Department of Pathology, Harvard Medical School and Brigham and Women's Hospital, Boston, MA, 02115, USA
[5]Department of Biochemistry and Biophysics, University of California San Francisco, San Francisco, CA, 94143, USA
[6]Center for Advanced Technology, University of California San Francisco, San Francisco, CA, 94143, USA
[7]Department of Physics, Simon Fraser University, Burnaby, BC V5A 1S6, Canada
[8]Helen Diller Family Comprehensive Cancer Center, San Francisco, CA, 94115, USA
[9]Chan Zuckerberg BioHub, University of California San Francisco, San Francisco, CA, 94143, USA
[10]Center for Cellular Construction, University of California San Francisco, San Francisco, CA, 94143, USA
[11]Apertura Gene Therapy, 345 Park Ave South, New York, NY 10010
[*]These authors contributed equally to this work
[†]mthomson@caltech.edu

## Abstract

**Gene regulatory networks within cells modulate the expression of the genome in response to signals and changing environmental conditions. Reconstructions of gene regulatory networks can reveal the information processing and control principles used by cells to maintain homeostasis and execute cell-state transitions. Here, we introduce a computational framework, D-SPIN, that generates quantitative models of gene regulatory networks from single-cell mRNA-seq datasets collected across thousands of distinct perturbation conditions. D-SPIN models the cell as a collection of interacting gene-expression programs, and constructs a probabilistic model to infer regulatory interactions between gene-expression programs and external perturbations. Using large Perturb-seq and drug-response datasets, we demonstrate that D-SPIN models reveal the organization of cellular pathways, sub-functions of macromolecular complexes, and the logic of cellular regulation of transcription, translation, metabolism, and protein degradation in response to gene knockdown perturbations. D-SPIN can also be applied to dissect drug response mechanisms in heterogeneous cell populations, elucidating how combinations of immunomodulatory drugs can induce novel cell states through additive recruitment of gene expression programs. D-SPIN provides a computational framework for constructing interpretable models of gene-regulatory networks to reveal principles of cellular information processing and physiological control.**

# Introduction

The human genome encodes more than $30,000$ genes, but a typical human cell expresses only 5,000 genes at any given time [1]. Gene regulatory networks within cells modulate gene expression based upon environmental cues and the cell's internal state [2, 3, 4, 5]. In gene regulatory networks, transcription factors and associated regulatory proteins interact with one another and gene promoters to activate or repress gene transcription [6, 7, 8]. Gene regulatory networks play a central role in cellular decision-making processes [9, 10, 11, 12]. However, we have a limited understanding of the global logic encoded within gene regulatory networks and the underlying principles of cellular information processing.

Global reconstruction of gene regulatory networks in *E. coli*, yeast, and sea urchin embryos have revealed features of biological information processing, including network modularity, recurring network motifs, and combinatorial logic through the assembly of transcription factor complexes at gene promoters [13, 14, 15, 16, 17, 18, 19]. However, our understanding of the information-processing principles of gene regulatory networks is based on a limited number of network reconstructions performed on model organisms over a limited range of physiological conditions. Even in *E. coli*, a large fraction of genes have unknown regulators [20]. In metazoans, gene network models have primarily focused on sub-circuits involved in specific processes like T-cell activation, T-cell fate selection, and embryonic stem cell differentiation [10, 11, 12]. The comparative analysis of gene regulatory network architectures across cell types, developmental stages of an organism, and species therefore remains nascent, and there are very few quantitative models of regulatory networks that can predict the response of a cell to signals, genetic perturbations, or therapeutics. Cells simultaneously regulate tens of different processes including transcription, metabolism, protein degradation, and differentiation. How gene regulatory networks globally modulate core cellular processes in parallel, in response to environmental conditions, remains poorly understood.

Historically, gene regulatory network reconstruction and modeling have been constrained by the number of biochemical or genetic measurements required to reconstruct networks with hundreds to thousands of interacting protein components. Classical biochemical approaches, for example, perform bottom-up gene network reconstructions through pairwise binding measurements requiring order-$M^2$ biochemical assays for a network with $M$ components [21, 22]. Genetic approaches use a top-down strategy that infers gene regulatory network architecture from perturbation-response experiments where genes are knocked down or protein activity is inhibited with small molecules. Perturbation of one network component alters gene expression for subsets of genes; through the top-down association of perturbations with responses, perturbation approaches enable the identification of gene regulators, the organization of regulators into pathways, and the construction of integrated network models [23, 24, 25, 26]. For global analysis of gene regulatory networks, however, perturbation approaches require knocking out hundreds to thousands of genes while monitoring transcription across thousands of genes. Historically, experimental realization of genome-wide network reconstruction through

genetic perturbation has, like biochemistry, been limited by experimental scale when genes were knocked out in bulk assays.

Developments in single-cell genomics and perturbation barcoding circumvent some of the conventional limitations of perturbation-driven network reconstruction. Perturbation barcoding approaches, including Perturb-seq, click-tags, and MULTI-seq, allow the transcriptional state of each cell in a cell population to be measured across thousands of different genetic, signaling, and small molecule conditions [27, 28, 29, 30]. These experimental approaches identify the perturbation delivered to every single cell in a population while also measuring changes in the entire transcriptome through single-cell mRNA-seq. Given that single-cell perturbation measurements allow interrogation of a cell population's response to thousands of genetic, signaling, or small molecule perturbations, a core challenge is developing computational methods to integrate data from thousands of such experiments into a gene regulatory network model that can classify perturbations, map the flow of information across a regulatory network, and predict cellular response to novel perturbations.

Here, we introduce a mathematical modeling and network inference framework, D-SPIN (Dimension-reduced Single-cell Perturbation Integration Network), that constructs gene regulatory network models directly from single-cell perturbation-response data. D-SPIN models the cell as a collection of interacting gene-expression programs and constructs a probabilistic model, known as a Markov random field or spin network, to infer regulatory interactions between gene-expression programs and applied perturbations. Spin network models originated in statistical physics and have been generalized and applied to a wide range of systems and applications through work on maximum entropy methods in physics and on associative memory of Hopfield networks in machine learning [31, 32]. Spin-network models have shown promise for the inference of gene regulatory network models from bulk gene expression data [33, 34] where application has focused on learning networks that can capture mean cell states through the original Hopfield memory storage algorithms. However, the application of spin network models in the single-cell context remains limited due to the absence of an effective statistical inference framework that can scale to large networks, incorporate perturbations, and capture the information contained in the entire cell state distribution measured in single-cell experiments. To enable information integration in large datasets with thousands of conditions and millions of single cells, D-SPIN exploits a natural factoring within the mathematical structure of Markov random fields inference to separate the learning problem into two steps: 1) construction of a unified gene regulatory network model and 2) inference of how each perturbation interacts with the gene programs within the unified network. To maximize the interpretability and computational efficiency of D-SPIN models, D-SPIN can operate on single genes or gene-expression programs, and the program-level description enables D-SPIN to derive a coarse-grained description of a cellular regulatory network in terms of physiological units—gene-expression programs—and their interactions with applied perturbations. D-SPIN can accommodate a wide range of different perturbation types including genetic perturbations, small molecules, signaling conditions, and even physiological states of health and disease. In benchmarking tests, we demonstrate that D-SPIN exhibits state-of-the-art performance on simulated gene network inference tasks for networks with thousands of elements.

We apply D-SPIN to construct gene regulatory network models from experimental datasets with thousands of perturbations and millions of cells, including two of the largest single-cell mRNA-seq datasets in existence: a genome-wide Perturb-seq experiment on the K562 Chronic Myelogenous Leukemia cell line [28] and a new human immune cell drug-response experiment we performed. The integrated regulatory network models reveal global organizing principles of cellular regulation and perturbation responses, including the organizing principles of cellular pathways and networks of pathways used to maintain homeostasis in K562 cells in response to perturbation of core cellular processes such as respiration, transcription, and translation. Applied to profile the response of human immune cells to combinations of immunomodulatory drugs, D-SPIN models demonstrate that drug combinations can generate novel cell states through additive recruitment or superposition of drug-specific gene expression programs. Broadly, D-SPIN provides a computational framework for large-scale modeling and comparative analysis of gene regulatory networks across cell types, physiological conditions, and organisms.

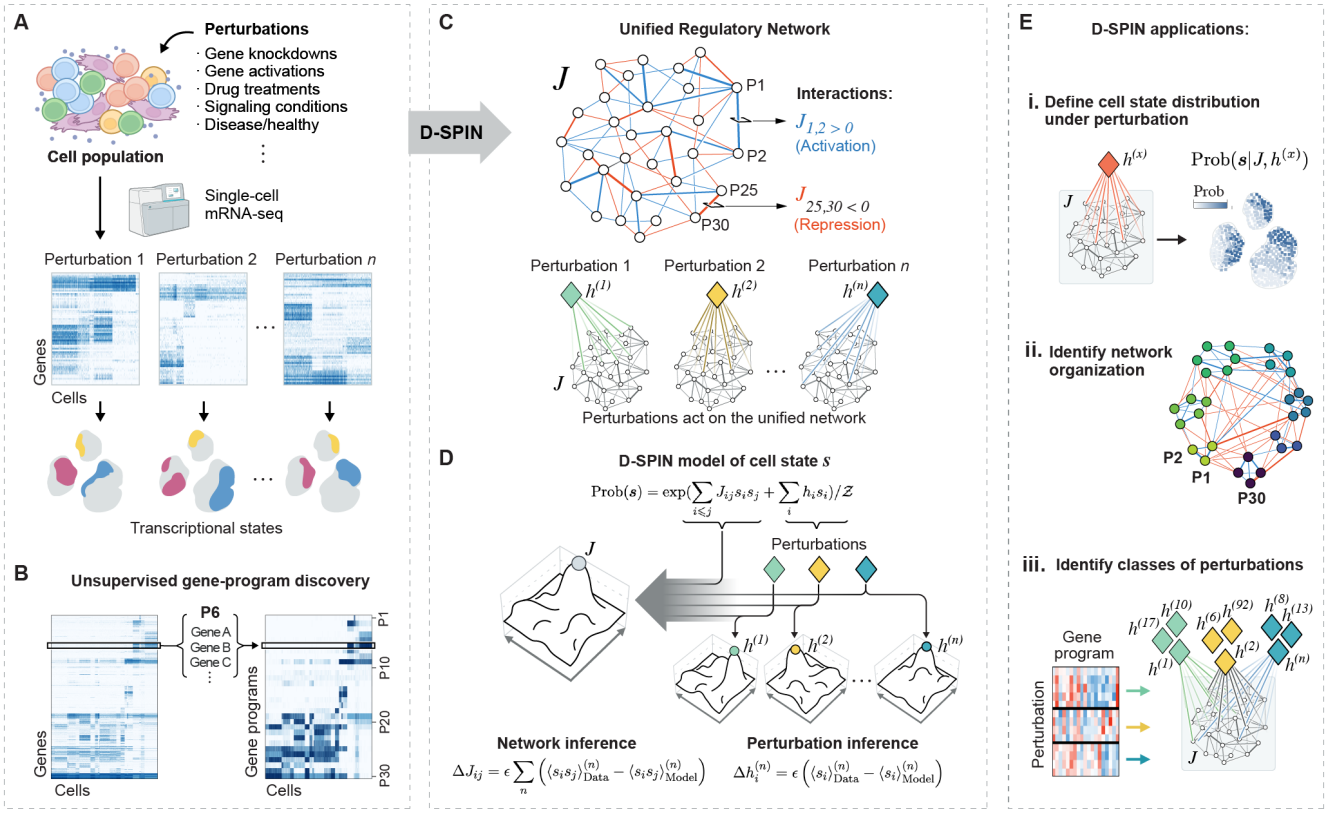


Figure 1: **D-SPIN constructs unified gene regulatory network models from single-cell mRNA-seq data collected across perturbation conditions.** (A) D-SPIN accepts as input single-cell mRNA-seq data from a cell population profiled across a series of different perturbed experimental conditions such as genetic perturbations, drug treatments, signaling conditions, or disease/healthy conditions. (B) To reduce the dimensionality of the network-inference task and enhance interpretability, D-SPIN extracts gene-expression programs from the data using orthogonal non-negative matrix factorization. D-SPIN is also capable of constructing gene-level networks with thousands of nodes using pseudolikelihood-based inference (STAR Methods 1.3). (C) D-SPIN then constructs a unified regulatory network $\boldsymbol{J}$, whose edges represent inferred interactions between gene-expression programs. D-SPIN also infers interactions between the network and applied perturbations. (D) Mathematically, D-SPIN constructs the regulatory network model by inferring a probabilistic model that encodes the probability of each transcriptional state in terms of an interaction matrix $\boldsymbol{J}$ and sample-specific perturbation response vectors $\boldsymbol{h}^{(n)}$. D-SPIN scales to large datasets with millions of cells through a parallelized, maximum-likelihood procedure. At each step of inference, D-SPIN generates model samples for each perturbation condition in parallel, and then uses the difference between model and data cross-correlation and mean to update the network weights $\boldsymbol{J}$ and each perturbation response vector $\boldsymbol{h}^{(n)}$. (E) D-SPIN yields a probabilistic model that can be interpreted as a gene regulatory network. The D-SPIN model can be applied to (i) estimate the distribution of transcriptional states defined by the regulatory network model under a specific condition, (ii) reveal principles of gene regulatory network organization, including modularity and sub-network architecture, and (iii) classify perturbations into groups that reflect biological function.

## D-SPIN infers a unified regulatory network model from single-cell perturbation data

Here, we develop mathematical and computational strategies to integrate data across many different single-cell perturbation experiments into a gene regulatory network model that provides qualitative and quantitative insights into the architecture of cellular pathways and regulatory strategies. D-SPIN solves the gene regulatory network inference problem by constructing a probabilistic model that encodes the distribution of transcriptional states within a cell population across a set of perturbed conditions (Figure 1). D-SPIN can be applied to model cell populations perturbed by inputs including gene knockdown/activation, altered signaling conditions, and disease states, as well as the addition of small molecules or other therapeutics. Mathematically, D-SPIN builds a spin network model or Markov random field, which can be interpreted as an interaction network. Spin-network models were initially introduced in physics to study magnetic materials known as spin glasses, whose properties emerge through positive and negative interactions between localized magnetic moments or spins [31, 32, 35]. In physics and machine learning, spin network models have been generalized and applied to many different systems of interacting elements, including neural networks, bird flocks, and proteins [36, 37, 31, 38, 33, 34], and also to study the storage of memories in spin networks known as Hopfield networks. The application of spin network models to single-cell data has been limited due to mathematical and computational challenges associated with the problem of solving the spin network inference problem on large datasets. D-SPIN exploits a factoring of the spin-network Hamiltonian to develop an efficient computational procedure for inferring spin-network models from large single-cell datasets across many perturbation conditions.

In general, spin network models consider a network of $M$ interacting elements $s_i$. The goal of the framework is to construct a model that encodes the probability $P(s_1, \ldots, s_M)$ of each potential configuration of the network of elements based upon inferred interactions between elements, as quantified in a coupling matrix $\boldsymbol{J}$. In D-SPIN, $s_i$ is chosen to be three-state discrete variables in $[-1, 0, 1]$ instead of two-state variable in $[-1, 1]$ as is common in statistical physics. The three-state representation is chosen to model intermediate levels of gene expression, which can be functionally relevant for gene regulation and for improved quantitative agreement between model and experimental cell state distributions (STAR Methods 1.2). Similar discretization methods have been deployed to reduce noise and accommodate nonlinear dependencies from the pioneering works in modeling gene expression networks using microarray measurements [39, 40, 41].

D-SPIN can be applied to model networks with thousands of individual genes. However, gene-level networks with thousands of nodes are difficult to interpret. Therefore, in addition to providing a scalable method for inferring D-SPIN models on individual genes (STAR Methods 1.3), we also designed D-SPIN to construct reduced- or low-dimensional spin network models by modeling interactions between gene programs, i.e., co-regulated groups of genes [42, 43, 44]. Cells regulate their transcriptional state by modulating transcription factors that impact sets of genes related to differentiation state, cellular pathways, and core physiological processes. By focusing on gene expression programs, rather than single genes, D-SPIN generates low-dimensional models of cellular regulation that are interpretable and can be inferred from data through a computationally efficient inference procedure. Building upon previously published work, D-SPIN contains an automatic procedure for identifying co-regulated groups of genes (gene programs) from single-cell data, based on unsupervised orthogonal nonnegative matrix factorization (oNMF) and phenotypic gene-program annotation (Figures 1A and 1B, STAR Methods 1.8) [45, 46, 47, 48], but can also accommodate pre-defined gene sets from prior biological knowledge (STAR Methods 1.9).

Following the extraction of gene-expression programs, D-SPIN applies a maximum likelihood statistical inference procedure to infer a network of regulatory interactions between gene programs and between programs and applied perturbations (Figure 1C). Regulatory interactions between gene programs are represented in an interaction matrix $\boldsymbol{J}$ where each entry $J_{ij}$ quantifies the inferred interaction between program $i$ and $j$. Interactions between the regulatory network and perturbations are encoded in a set of perturbation response vectors $\boldsymbol{h}^{(n)}$ where $h_i^{(n)}$ quantifies the interaction of the perturbation $n$ with gene program $i$.

Mathematically, an energy function $E(\boldsymbol{s})$ computes the effective energy of a given transcriptional state $\boldsymbol{s}$ by balancing the influence of the regulatory network (encoded in $\boldsymbol{J}$) and the perturbation-specific inputs (encoded in each $\boldsymbol{h}^{(n)}$). The resulting probability of a given transcriptional state $\boldsymbol{s}$ is $P(\boldsymbol{s}) =$

$\frac{1}{Z}\exp[-E(\boldsymbol{s};\boldsymbol{J},\boldsymbol{h}^{(n)})]$, where

$$E(\boldsymbol{s};\boldsymbol{J},\boldsymbol{h}^{(n)}) = -\sum_{ij} J_{ij}s_is_j - \sum_i h_i^{(n)}s_i \tag{1}$$

and $Z$ is a normalizing constant ensuring all probabilities sum to 1 (STAR Methods 1.2) [49, 50]. The regulatory network accounts for activating/positive ($J_{ij} > 0$) and inhibitory/negative ($J_{ij} < 0$) interactions between gene programs. Perturbations exert a bias on each gene program to increase the likelihood of being on ($h_i > 0$) or off ($h_i < 0$).

Given single-cell gene-expression data collected from a series of perturbation conditions, D-SPIN performs an optimization procedure through gradient ascent to estimate model parameters $\boldsymbol{J}$ and $\boldsymbol{h}^{(n)}$ (Figure 1D). For spin network models, the optimization procedure yields closed-form parameter-update rules

$$\Delta J_{ij} = \epsilon \sum_n \left( \langle s_is_j\rangle_{\text{Data}}^{(n)} - \langle s_is_j\rangle_{\text{Model}}^{(n)} \right) \tag{2}$$

$$\Delta h_i^{(n)} = \epsilon \left( \langle s_i\rangle_{\text{Data}}^{(n)} - \langle s_i\rangle_{\text{Model}}^{(n)} \right) \tag{3}$$

where $\langle s_is_j\rangle_{\text{Data}}^{(n)}$ are gene-program cross-correlations computed from data for each perturbation condition $n$, $\langle s_is_j\rangle_{\text{Model}}^{(n)}$ are gene-program cross-correlations computed from the model, and $\epsilon$ is a learning rate (STAR Methods 1.3 and 2.4). At each step of model inference, the procedure samples from the model for each perturbation condition and updates the regulatory network $\boldsymbol{J}$ and each perturbation vector $\boldsymbol{h}^{(n)}$. In general, spin network model inference for a single condition is a convex optimization problem, and we show in the supporting materials that the multi-condition optimization problem solved by D-SPIN remains convex, so that the D-SPIN optimization has a single unique solution (STAR Methods 1.5). As a statistical model, Fisher information can be applied to quantify the uncertainty of edges in the D-SPIN model to guide interpretation and experimental design (STAR Methods 1.6) [51]. The D-SPIN inference procedure solves the full maximum likelihood inference problem associated with spin-network inference. Therefore, D-SPIN inference is distinct from the conventional Hebbian learning rule developed to encode memories in Hopfield networks, as applied in a pioneering study to perform regulatory network inference from bulk microarray data [31, 33].

The form of the spin network model learning rules enables us to develop a highly efficient and scalable training algorithm for D-SPIN by exploiting intrinsic parallelization. Training of general graphical models requires estimating the gradient function for every data point separately, which is typically computationally extensive. In contrast, in D-SPIN the gradient is pooled for each experimental condition, and only depends on the mean $\langle s_i\rangle$ and cross-correlation $\langle s_is_j\rangle$ between gene programs in each condition. Consequently, the training of the network can be deployed in parallel, with each computational thread computing the gradient based on the mean and cross-correlation of a single experimental condition. This approach eliminates the need to estimate gradients on every data point separately (i.e., every cell's transcriptional state) and minimizes data communication cost by only requiring the mean and cross-correlation to be exchanged during parallelization. Therefore, we routinely use hundreds of CPU cores for the optimization, enabling efficient network inference over large datasets. Practically, to avoid overfitting on a large number of similar conditions, we estimate the network using a subset of experimental conditions with balanced samples from different phenomenological condition groups and utilize the inferred network to identify response vectors of each condition independently (STAR Methods 1.1 and 2.4).

Following inference of $\boldsymbol{J}$ and $\boldsymbol{h}$, D-SPIN models can be analyzed as network models (undirected graphical models) that reveal regulatory connections between cellular processes and identify interactions between applied perturbations and cellular gene-expression programs (Figure 1E). D-SPIN models can also be used to generate the distribution of transcriptional states in a cell population under normal and perturbed conditions and to explain how interactions within a gene network lead to a specific distribution of cell states in an experimental population. The models provide a computational strategy that can bridge geometric, cell clustering approaches like Uniform Manifold Approximation and Projection (UMAP) with mechanistic modeling approaches that seek to interpret changes in cell population structure in terms of underlying regulatory interactions. Geometric embedding approaches like UMAP, or even PCA, can help identify shifts in gene expression distributions within a cell population across conditions; however, interpretation of such shifts in terms of genes and gene interactions remains a major challenge [52]. D-SPIN provides a quantitative frame-

work that can reveal how perturbations to specific nodes of a network alter the distribution of transcriptional states in a cell population and, thus, alter the rendering of a cell population in linear or non-linear embedding methods like PCA and UMAP, respectively.

In addition to the gene-program level description, we also provide an optimized inference algorithm for building gene-level networks where we scale inference to networks that contain thousands of genes using the concept of pseudolikelihood [53, 54, 55, 56]. Rather than performing inference using the global likelihood function constrained by pairwise correlations, pseudolikelihood methods consider the conditional probability distribution, $P(s_k|s_1, \ldots, s_{k-1}, s_{k+1}, \ldots, s_M)$ for a single network component $s_k$ given data about all other network components excluding $s_k$. Parameter inference is performed by maximizing the logarithm of a pseudolikelihood function, constructed as the product of $M$ single spin conditional likelihood functions, yielding a regression-like problem. As an approximation to the likelihood function, pseudolikelihood methods are computationally efficient and scalable. Model parameters estimated by pseudolikelihood converge to the ground truth, with error decreasing exponentially with sample size in theoretical analysis and numerical tests on classic spin network models from physics [54, 55]. Additionally, the framework achieves high accuracy in cases where networks have strong interactions with local fields, and the number of data points required to estimate network parameters scales logarithmically with the number of network nodes [55]. Pseudolikelihood methods can also be applied to estimate directed interactions [54, 56]. Pseudolikelihood-based inference methods enable scaling of D-SPIN to infer gene-level models in gene regulatory networks with thousands of nodes.

## D-SPIN achieves state-of-the-art performance on gene regulatory network inference

To evaluate the accuracy of D-SPIN in performing network inference, we first applied D-SPIN to reconstruct and analyze a model of the hematopoietic stem cell (HSC) differentiation network using the network simulation and benchmarking framework BEELINE [57]. The HSC network has two major groups of regulatory genes. The first group includes the transcription factor Pu1, controlling granulocyte/monocyte differentiation, and the second group includes Gata1, controlling megakaryocyte/erythrocyte differentiation [58]. BEELINE simulates single-cell transcription data from biologically identified regulatory networks by generating and simulating ordinary differential equation (ODE) models with stochastic noise. BEELINE also includes existing network inference methods for benchmark accuracy evaluation. We used the BEELINE framework to generate synthetic gene-expression profiles from the HSC network for 22 perturbation conditions, encompassing knockdown/activation of each network node individually by turning down/up the transcription rate (Figures 2B and 2C, STAR Methods 2.6). For interpretation of the data we used the known biological functions of each transcription factor to classify the single-cell expression profiles into 7 different cell state designations, including monocytes (Mono), granulocytes (Gran), erythrocytes (Ery) and megakaryocytes (Mega) [58].

We applied D-SPIN to reconstruct a gene regulatory network model from the simulated data and compared the accuracy of the D-SPIN model to 3 different network reconstruction methods (PIDC, GRNBoost2, and GENIE3) that were the top performers in the BEELINE benchmarking study [57]. We applied D-SPIN to generate a single network model encoded in the gene interaction matrix $\boldsymbol{J}$ and generate perturbation vectors $\boldsymbol{h}^{(n)}$ for each perturbation condition (Figure 2D, Figure S2A). We applied the algorithms to individual genes, rather than gene programs, given the relatively compact size (11 nodes) of the HSC network used in the BEELINE benchmarking evaluation. Each method generates a set of inferred edges and provides confidence/importance estimates for each edge. We compared the methods by considering the accuracy of the top 10 and top 20 edges identified by each method.

On the simulated network reconstruction task, D-SPIN consistently achieved higher performance than comparable methods, primarily due to the natural way in which D-SPIN can accommodate perturbation data. For D-SPIN, on average 0.96 of the top 10 edges found by the model were correct across inference runs, as compared to 0.6, 0.77, and 0.7 for PIDC, GRNBoost2, and GENIE3, respectively (Figure 2E). For D-SPIN, 0.75 of the top 20 edges were correct, compared with 0.625, 0.645, and 0.665 for PIDC, GRNBoost2, and GENIE3, respectively (Figure 2E).The inference algorithm of D-SPIN is crucial to its accuracy improvement, as the Hebbian learning rule in Hopfield networks [31, 33] only has 0.55 accuracy on the top 20 edges (SI Table 2). In one example of D-SPIN outperforming existing methods, the networks inferred by PIDC, GRNBoost2, and GENIE3 all identified strong spurious activating interaction inside the Mega-Ery gene

module. For example, Scl-Fog1 is predicted as one of the strongest edges in PIDC, GRNBoost2 and GENIE3 with weights 1.17, 1.34, and 1.70, while D-SPIN assigns an average weight of 0.35 to the edge and ranks it 24th. Similarly, Scl-Eklf is ranked within the top-12 edges in other methods with weights 0.99, 0.76, and 0.71, while D-SPIN assigns weight 0.4 to the edge and ranks it 22nd. D-SPIN reveals the central regulatory role of Gata1 by significantly reducing the weights of these false positive edges. D-SPIN also removed spurious inhibitory interactions between the two major groups of genes (Mono-Gran module and Mega-Ery module) from the edge predictions, e.g., Pu1-Eklf or Pu1-Fli1, revealing the actual regulatory target of Pu1. When spurious edges occurred, they were similar across the different inference frameworks: all four methods identified spurious edges in strongly connected regions of the HSC network, struggling to separate direct versus indirect connections. However, for D-SPIN uniquely, spurious edges can be further reduced through the sampling of higher-order perturbations. By including additional 96 double-gene perturbations, D-SPIN achieves 100% accuracy on the top 10 edges and improves the accuracy of the top 20 edges from 0.75 to 0.81 ("D-SPIN_Comb" in Figure S2B(i), SI Data).

Additionally, D-SPIN also achieves superior accuracy in inferring a directed network model using pseudolikelihood (Figure S2B(ii)). The directed HSC network has 26 edges; the top 26 predicted edges of D-SPIN have an accuracy of 0.73, outperforming the other three methods (PIDC, GRNBoost2, and GENIE3) with accuracies of 0.41, 0.47, and 0.53, respectively. Despite the strength of D-SPIN in directed network inference, we focus our analysis on undirected networks due to the inherent challenge in defining consistent directed network models using steady-state cell state distributions (STAR Methods 1.3) [59].

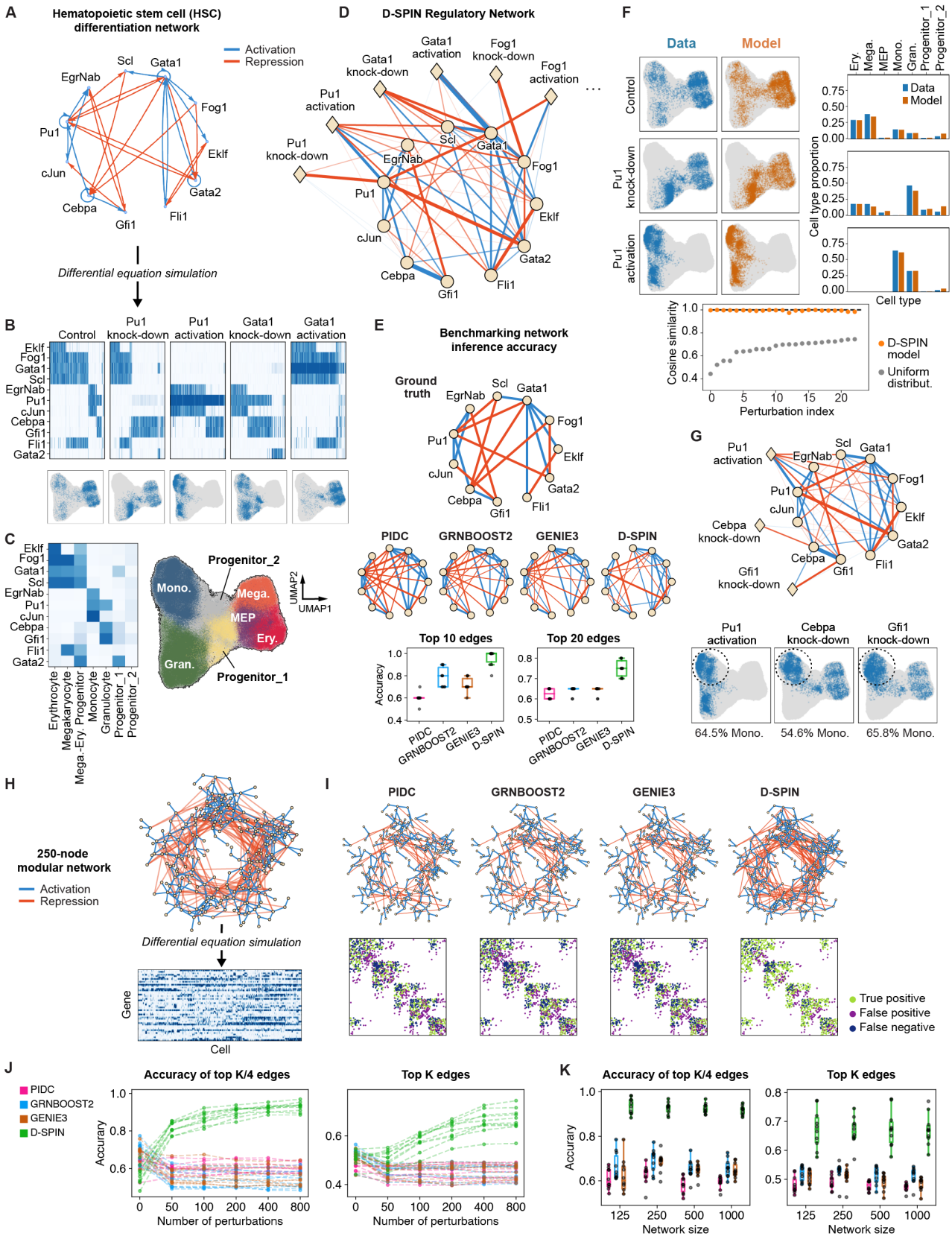


Figure 2: Caption on the following page

Figure 2 **D-SPIN achieves state-of-the-art performance on network inference benchmarking tasks and reveals network-level mechanisms of cell fate modulation by transcription factor perturbations.** (A) HSC regulatory network model [58] contains 11 transcription factors (TFs) that interact through activation (blue) and repression (red) to modulate the differentiation of HSCs into different cell types. (B) Example simulated gene-expression profile heatmaps generated using a published computational framework BEELINE from the network across a series of simulated single-gene knockdown and activation conditions. (C) Heatmap shows average TF expression of 7 clusters of cell states generated across all simulated conditions. Each cluster is annotated with its TF expression pattern based on the biological context of the HSC network model [58]. All simulated cell states and clusters are visualized in the UMAP embedding. (D) Network diagram shows the inferred unified regulatory network model by D-SPIN that encodes inferred interactions between TFs as an interaction graph. Edges with diamond markers in the network show D-SPIN inferred perturbation vectors that estimate how knockdown or activation of TFs (e.g. Gata1 knockdown) impacts the regulatory network through up- or downregulation of TFs. (E) Diagram of the true network and inferred regulatory networks by D-SPIN and other state-of-the-art methods including PIDC, GRNBoost2, and GENIE3 [57]. The inference accuracy is quantified by the box plot of the accuracy of the top 10 and 20 inferred edges across 10 random repeats of parameters in differential equation simulation using BEELINE. The ground truth network has a total of 20 edges. (F) (left) UMAP embedding comparisons of simulated single-cell data by BEELINE and state distributions generated by D-SPIN models for control and Pu1 activation/knockdown. (right) Bar plots quantify the proportion of HSC-derived cell types in each condition from simulated data and D-SPIN models. (Bottom) Cosine similarity between cell state distributions of data and model. D-SPIN models generate cell-state distribution highly consistent with simulated data by applying perturbations to the underlying network. (G) (top) Network diagram and (bottom) UMAP embedding of three different perturbations that generate increased monocyte population. The D-SPIN model reveals network edges that mediate the response of the network to different perturbations through the application of a formal reasoning framework to the regulatory network model. Both Pu1 activation and knockdown of Cebpa or Gfi1 generate increased monocyte states by directly interacting with monocyte genes or indirectly through the interaction between Gfi1 and EgrNab. (H) (top) Network diagram of an example 250-node modular network with 5 mutually inhibiting modules and (bottom) simulated single-cell data from the network for evaluating the accuracy and scalability of network inference. (I) Diagrams of the inferred network of different methods as in (top) subnetwork of correct-inferred edges and (bottom) adjacency matrices with true positives, false positives, and false negatives. D-SPIN achieves superior accuracy compared to existing methods (J) The network inference accuracies are plotted with the number of perturbations in 10 randomly generated modular networks. The accuracy of D-SPIN continuously increases with perturbation number while other methods do not improve or slightly decrease. (K) Bar plots quantify the network inference accuracy of modular networks with different sizes of 125, 250, 500, and 1000 under 800 perturbations in 10 random repeats. D-SPIN achieves significantly increased accuracy compared with other methods across network sizes.

## D-SPIN constructs an interpretable, generative model of HSC perturbation response

D-SPIN is a probabilistic, generative model that can generate the distribution of transcriptional states in a cell population subjected to a perturbation. In contrast, most existing network inference methods are not generative and use either regression or information-theoretic measures to identify and list candidate gene interactions [57]. For the HSC network, the distribution generated by the D-SPIN model is highly concordant with the data distribution across all perturbation conditions, as visualized by the UMAP embedding and cell state distribution comparison (Figure 2F, Figure S2C). Quantitatively, the cell state distributions by D-SPIN are all above 96% cosine similarity with the simulated data. The quantitative agreement is especially remarkable as D-SPIN is a minimal and interpretable model that encodes each network interaction with a single signed strength parameter. Thus, for the HSC network, D-SPIN constructs a quantitative model of how the cell population is generated by the underlying gene regulatory network. We can apply the model to compute quantitative statements of gene expression change induced by each perturbation and to analyze the role of potential gene-gene interactions in mediating the perturbation response of the regulatory network.

As generative probabilistic models, D-SPIN models can be analyzed to gain insight into how edges within a D-SPIN network mediate the response to applied perturbations. We developed a formal framework

to determine how the knockdown or activation of specific nodes in the network might impact the distribution of cell states generated by the regulatory network and to define how individual edges contribute to the perturbation response of the network (STAR Methods 2.7). Specifically, for a perturbation $\boldsymbol{h}$, we define the marginal activity $\hat{h}_k$ of each gene $k$ to be the ratio between the marginal probability of the node $k$ being expressed $P(s_k = 1 \text{ given } \boldsymbol{h})$ or silenced $P(s_k = -1 \text{ given } \boldsymbol{h})$. Then we evaluate the influence of every edge $J_{ij}$ in the network on $\hat{h}_k$ to generate an edge sensitivity score $\epsilon(J_{ij}, k)$, which is the change of marginal activity $\hat{h}_k$ before and after hypothetical deletion of edge $J_{ij}$ from the network model. The edge sensitivity score allows us to rank the importance of network edges in modulating the response of the network to the knockdown or activation of a specific gene.

For example, in both the D-SPIN model and training data, the knockdown of the transcription factor Pu1 results in a complete loss of monocyte cell states and an increase in granulocyte cell states (Figure 2F). The formal reasoning framework can provide insight into network edges that mediate the response of the network to Pu1 knockdown and the resulting change in population structure. According to the node marginal distribution and edge sensitivity analysis, under Pu1 knockdown the marginal activity of EgrNab and cJun decrease by $-12.9$ and $-6.4$. This is primarily induced by the positive interactions of Pu1-EgrNab and Pu1-cJun with the edge sensitivity score $\epsilon$(Pu1-EgrNab, EgrNab) $= -4.4$ and $\epsilon$(Pu1-cJun, cJun) $= -3.2$ (Figure S2D(i)). Pu1, EgrNab, and cJun are expressed in the monocyte state [58], and the repression of these genes corresponds to the loss of the monocyte cell state. Conversely, under Pu1 knockdown, Gfi1 and Cebpa, genes expressed in granulocyte cell state [58], have an increased marginal activity by 3.0 and 2.9 respectively, corresponding to the increased granulocyte cell states. Edge sensitivity scores show that the expression of Gfi1 and Cebpa is impacted both by the positive interactions of Pu1-Gfi1 and Pu1-Cebpa and by the negative interaction of Gfi1-EgrNab. For Gfi1, Gfi1-EgrNab has a stronger impact of $\epsilon$(Gfi1-EgrNab, Gfi1) $= 4.7$ compared with $\epsilon$(Pu1-Gfi1, Gfi1) $= -2.6$. The response of Cebpa is also primarily induced by the edge Gfi1-EgrNab with $\epsilon$(Gfi1-EgrNab, EgrNab) $= 4.3$ compared with $\epsilon$(Pu1-EgrNab, EgrNab) $= -3.9$ (Figure S2D(ii)(iii)). The Mega-Ery module (Eklf, Fog1, Gata1, Scl, and Fli1) is affected by both the negative interaction with Pu1 (which is downregulated) and Cebpa (which is unregulated), and the overall marginal activities exhibit weak inhibition $(-0.7, -1.5, -2.6, -1.3, -1.4)$.

As another example of formal reasoning, in both simulated data and the D-SPIN model, both Pu1 activation and the knockdown of either Cebpa or Gfi1 individually lead to an increased prevalence of monocyte cell states, but with different inferred network mechanisms (Figure 2G). Activating Pu1 causes the increase in marginal activity for other monocyte genes, EgrNab and cJun, by 2.4 and 2.8 which is mediated by the network edges Pu1-EgrNab and Pu1-cJun, with edge sensitivity scores of 4.4 and 3.1. Alternately, Gfi1 has both positive and negative interactions with monocyte genes (Pu1, EgrNab, and cJun), and edge sensitivity analysis on Pu1 under the Gfi1 knockdown condition shows that the negative edge Gfi1-EgrNab has a stronger effect on Pu1 with score $\epsilon$(Gfi1-EgrNab, Pu1) $= 2.8$ compared with the positive edge Gfi1-Pu1 with a score $\epsilon$(Gfi1-Pu1, Pu1) $= -2.6$. Moreover, the marginal activities of monocyte genes Pu1, EgrNab, and cJun under Gfi1 knockdown increase by 3.0, 2.4, and 2.7, leading to an increased monocyte population (Figure S2D(iv)). Similarly, under Cebpa knockdown, the marginal activity of Gfi1 is strongly decreased by $-6.4$, which is primarily due to the interaction of Cebpa-Gfi1 with an edge sensitivity score of $-6.3$, therefore leading to a similar cell state change as in Gfi1 knockdown (Figure S2D(v)). Pu1 activation induces increased monocyte states through the positive interactions Pu1-EgrNab and Pu1-cJun, while increased monocyte states in Cebpa or Gfi1 knockdown are mediated by the negative edge Gfi-EgrNab. Therefore similar cell-state distribution change of increased monocyte states can be achieved by two different network-level mechanisms in the D-SPIN model.

Thus, D-SPIN, using simulated data, can infer an accurate and generative regulatory network model of a single-cell population by integrating information across perturbation conditions. The D-SPIN model can be analyzed through formal reasoning—as in this example of tracking monocyte and granulocyte cell state distributions—to hypothesize network edges and network-level mechanisms that might be important for determining how the structure of a cell population shifts when individual network nodes are perturbed. The D-SPIN model demonstrates how activation of Pu1 and knockdown of Cepba can engage two different sets of network edges to increase the prevalence of monocyte cell states. The inferred regulatory network reveals functional interactions between regulatory genes and provides a map of how the distribution of cell states in the underlying HSC cell population is generated through interactions between internal regulators.

## D-SPIN achieves high accuracy in large-scale regulatory network inference

(LargeNetwork-T1) To demonstrate the versatility of D-SPIN, we expanded the validation to infer the structure of large-scale simulated networks with hundreds to a thousand nodes, where we performed network inference using pseudolikelihood-based methods (STAR Methods 1.3). We explored two distinct network architectures: highly modular networks and Erdős–Rényi (ER) random networks (Figure 2H, Figure S2E). The modular networks are characterized by activating interactions between genes inside the same module, and inhibitory interactions between neighboring modules (Figure 2H). Modular networks contain key structural properties of biological gene regulatory networks, where modules of coexpressed genes often perform coherent functions; further, modular networks present challenges for network inference procedures because gene coexpression makes it difficult to define the precise regulators that control module activation and inhibition [17, 43, 60, 51]. In ER networks, each gene has the same probability of interacting with any other gene (Figure S2E); ER networks are considered the most general statistical model for generating unstructured random networks [61, 62] and are often used for benchmarking inference methods.

Theoretical results [51, 63, 64] demonstrate that the inference of biological regulatory network models can be complicated by parameter degeneracy, where a set of network models can be consistent with a single set of measurements. Perturbations can be applied to break such degeneracy by driving a regulatory network into configurations that expose differences between network architectures that might generate very similar cell state distributions in an unperturbed context [51]. Unlike many existing gene regulatory network inference methods, D-SPIN can naturally integrate and synthesize information collected across many perturbation experiments into a single network model, so that D-SPIN can exploit the information provided by perturbation experiments to increase the accuracy of model inference. As perturbation design for large networks remains an open challenge, we applied random perturbations to each network, where the perturbations are independent, normally distributed random variables on each network node (STAR Methods 2.8).

In numerical tests, we found that D-SPIN's ability to integrate information from perturbations allows the algorithm to surpass the accuracy of other inference methods and to achieve increasing accuracy with increasing numbers of network perturbation experiments (Figure 2I, Figure S2F). The accuracy of network inference is measured by the accuracy of top $K$ edge predictions, where $K$ is the number of edges in the ground truth network. Without perturbation, the top $K$ edge accuracy of D-SPIN is around 0.5, similar to existing methods such as PIDC, GRNBoost2, and GENIE3. However, when perturbation conditions are included, D-SPIN's accuracy increases while the accuracy of other methods remains constant. By including 800 normally distributed, random perturbations, the accuracy of top $K$ predictions rises from 0.51 to 0.67 on modular networks (Figure 2J), and 0.40 to 0.70 on random networks (Figure S2G). More significantly, the accuracy of top $K/4$ predictions rises from 0.57 to 0.93 on modular networks and from 0.59 to 0.93 on random networks. In contrast to the increased inference accuracy with perturbation number for D-SPIN, applying the alternative network inference methods (PIDC, GRNBoost2, GENIE3) to each perturbation condition and averaging the results does not increase the accuracy of inference and even slightly decreases the accuracy (Figure 2J, Figure S2G).

We also noticed that in both types of networks, existing methods have much lower accuracy on negative edges. For example, in the modular networks, the average accuracy of PIDC, GRNBoost2, and GENIE3 are all 0.53 on positive interactions, but are 0.37, 0.35, and 0.36 respectively on negative interactions. In D-SPIN with perturbations, the average accuracy is 0.7 on positive edges and 0.8 on negative edges. These results suggest that inhibitory interactions are more difficult to identify and require combining information from different perturbations to discover [51].

Furthermore, the superior accuracy of D-SPIN scales to networks with up to a thousand genes, the largest-sized network that we could test within BEELINE. We performed a similar edge accuracy analysis on modular networks and random networks with sizes of 125, 250, 500, and 1,000 genes, and observed significant accuracy superiority of D-SPIN compared with other methods. On 1,000-node modular networks, the average top $K/4$ and top $K$ accuracy of D-SPIN is 0.92 and 0.67, while for PIDC, GRNBoost, and GENIE3, respectively, the top $K/4$ accuracy are 0.59, 0.66, 0.65, and the top $K$ accuracy are 0.47, 0.51, 0.48 (Figure 2K). On 1,000-node random networks, the average top $K/4$ and top $K$ accuracy of D-SPIN is 0.93 and 0.70, while for the other three methods, the top $K/4$ accuracy are 0.58, 0.62, 0.60, and the top $K$ accuracy are 0.38, 0.40, 0.38, respectively (Figure S2H).

## D-SPIN exhibits linear scaling of computation time with cell number

(LargeNetwork-T2) The unique model architecture and training algorithm of D-SPIN enables it to efficiently process large datasets containing hundreds of thousands of cells. The inference in D-SPIN is formulated as an optimization problem and can be solved in batches, thus circumventing the need to handle large matrix operations; as a result, the computation time of D-SPIN only increases linearly with the number of cells. We accessed the scalability of different network inference methods with the number of cells (Figure S2I, STAR Methods 2.9). We evaluated the running time of D-SPIN, PIDC, GRNBoost2, and GENIE3 with 250-node Erdős–Rényi (ER) random networks, with cell numbers ranging from 250 to 256,000. While PIDC initially outpaced other methods at lower cell numbers, D-SPIN became the fastest method for datasets exceeding 16,000 cells, a number most current single-cell studies surpass. Remarkably, at 256,000 cells, D-SPIN finished in 6 hours on 2 CPU cores, while all other methods could not finish within a week.

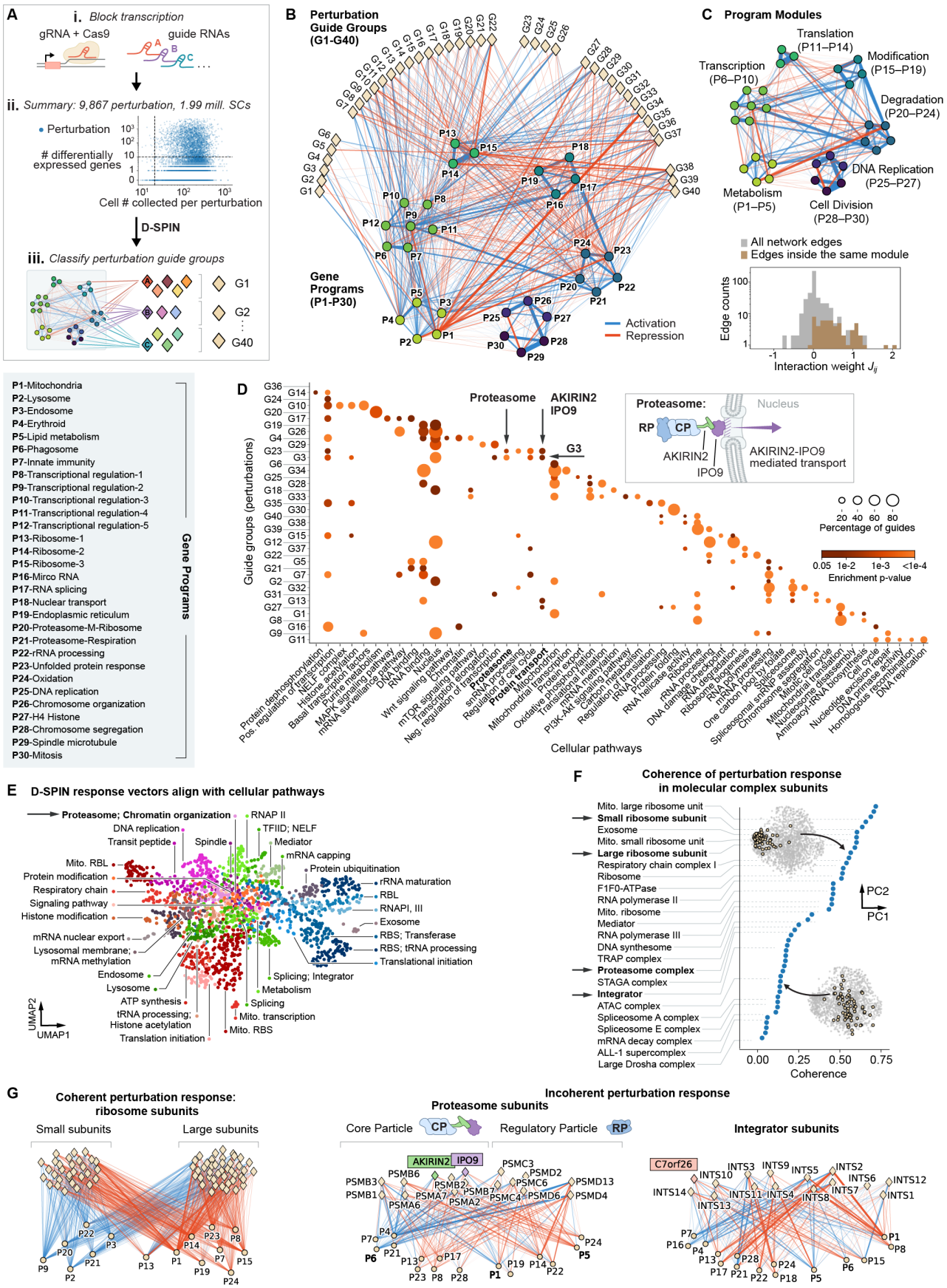


Figure 3: Caption on the following page

Figure 3 **D-SPIN generates a regulatory network model from genome-wide Perturb-seq data that reveals the architecture of cellular pathways and potential sub-functions for molecular complex subunits.** (A) Schematics of the experiments of the genome-wide Perturb-seq dataset [28]. (i) 9,867 gene knockdown perturbations are individually delivered to ∼2M human K562 cells using guide RNAs. (ii) Scatter plots of the number of differentially expressed genes (DEGs) and the number of cells that passed quality filtering for each perturbation. D-SPIN model construction was applied to 3,136 perturbations that induced more than 10 DEGs with more than 20 cells. (iii) The D-SPIN model constructs a unified regulatory network model and inferred interactions between each perturbation and the network, which classify gene-knockdown perturbations into groups. (B) Network diagram shows the inferred regulatory network model ($J$ matrix) between 30 gene-expression programs (P1-P30, circles), as well as the interactions between programs and 40 groups of gene-knockdown perturbations (G1-G40, diamond) encoded in response vectors ($h$ vectors). Interactions are rendered as positive (blue) or negative (red) edges with thickness scales with the strength of interactions. (left box) Gene programs are functionally annotated through gene ontology annotation tools [65, 66, 67] and manual lookup. (C) (top) Network diagram of the regulatory network model by D-SPIN exhibits modular structure with tightly-connected subnetworks. The 7 network modules can be assigned physiological functions based on the gene programs they encompass. (bottom) The histogram quantifies the distribution of all network edges ($J$ matrix entries) and edges inside the same module. Edges inside modules are mostly positive interactions and contain the majority of strong positive interactions in the network. (D) Bubble plot shows enriched cellular pathways in gene ontology and KEGG pathway from databases [65] for each guide RNA group where bubble size represents the percentage of enriched guides in the group. Gene targets in the same guide group identified by D-SPIN are involved in similar pathways or potentially have interactions, revealing associations between pathways, such as (arrow pointer)protein transport genes AKIRIN2 and IPO9 that were recently found to (inset schematics) mediate the import of proteasome into the nucleus [68]. (E) UMAP embedding of response vectors of each guide RNA where guides are colored by the major biological function that they are associated with. The UMAP is partitioned into regions corresponding to major cellular pathways including transcription, translation, respiration, and cell cycle. (F) The panel plots the regulatory coherence of macromolecular complex subunits, quantified by the average cosine similarity between the perturbation response vectors over all pairs of subunits. In high-coherence complexes, knockdowns of different subunits have similar interactions with the regulatory network encoded by the response vectors, as (top inset) visualized for an example complex in the PCA space of all response vectors. In low-coherence complexes, knockdowns of different subunits have different interactions with the regulatory network, as (bottom inset) shown in PCA for an example complex. The incoherence suggests distinct functions for complex subunits, (G) Diagrams of response vectors for subunits of the ribosome, proteasome, and integrator complexes, where diamond markers are targets of gene knockdowns and circle markers are gene programs of perturbation response. (left) The ribosome has a coherent signature where all subunits have similar inferred interactions with gene programs as encoded in response vectors, while (right) the proteasome and integrator each have signatures of sub-functionalization. In the example of proteasome, low coherence is observed as only a subset of subunits have an interaction on a gene program (P5, P6), or even display opposite interactions on a gene program (P1). The subunit partitions based on response vectors agree with the known structural partitions of the complexes, such as core and regulatory particles of the proteasome. Recently discovered genes AKIRIN2, IPO9, and C7orf26 are correctly grouped with the complex components that they interact with.

## Constructing a regulatory network model from genome-wide Perturb-seq data

Classic research in genetics has demonstrated that gene-knockdown experiments can be used to reconstruct the architecture of gene regulatory networks by observing how the knockdown of individual genes impacts the expression or activity of other genes [23, 24, 25, 26]. Perturb-seq is a single-cell genomics technique that enables highly parallelized genetic screens by combining the knockdown of individual genes using CRISPR/Cas9 with a transcriptome-scale single-cell mRNA-seq readout [27, 28]. Perturb-seq experiments can interrogate the impact of thousands of distinct gene knockdowns on cellular gene expression in parallel. D-SPIN provides a computational framework for the integration of data from thousands of perturbation experiments into a unified regulatory network model to generate insights into the architecture of cellular information flow and interactions between pathways.

We used D-SPIN to construct a gene regulatory network model using data from a genome-wide Perturb-seq experiment that used CRISPRi to knock down genes in the human chronic myelogenous leukemia (CML)(K562) cell line [28]. K562 is an erythrocyte precursor cell line with a BCR-ABL mutation, derived from a CML patient [69, 70]. In the genome-wide Perturb-seq experiment, $9,867$ genes were knocked down individually across 2 million single cells using CRISPRi. Prior to D-SPIN analysis, we pre-processed the data to identify guide RNAs that have an impact on the K562-cell transcriptional state (STAR Methods 2.1). Guide RNA perturbation impact varies (Figure 3A): 70.5% of perturbations had less than 10 differentially expressed genes (DEGs), 14.7% had 10-100 DEGs, and 14.8% had more than 100 DEGs. There is also great variation in the number of collected cells: 20.3% of perturbations had less than 100 cells collected.

To assess the network reconstruction quality of D-SPIN, we compared the network inference with independent biological measurements from ChIP-seq databases [71, 72, 73]. Chip-seq data measure the binding of transcription factors (TFs) to target gene promoters and have been used in the literature to assess the quality of inferred gene regulatory network models. We applied a gene selection procedure from the BEELINE benchmarking framework [57] where we selected two gene subsets TFs+500 and TFs+1000 that contain filtered TFs and the top 500 or 1000 highly variable genes, respectively (STAR Methods 2.10). We reconstructed the single-gene regulatory networks and compared the networks with TF-target regulation data obtained from ChIP-seq databases (Figure S3E, STAR Methods 2.10). The correspondence between inferred networks and ChIP-seq data is quantified by the early precision rate (EPR), which measures the proportion of inferred edges that can be associated with a TF binding event detected by ChIP-seq among the top $K$ predictions, relative to the expectation of a random predictor. $K$ is set to the total number of TF-target interactions in the ChIP-seq database. D-SPIN achieved leading correspondence with ChIP-seq data compared with other methods, with top $K$ EPRs of 4.48 and 4.36 for the TFs+500 and TFs+1000 datasets, respectively, compared to 2.88 and 2.68 in PIDC, 3.58 and 3.92 in GRNBoost2, and 3.39 and 3.13 in GENIE3.

Interpreting a single-gene network with hundreds or thousands of nodes can be complex. To increase the interpretability of the D-SPIN model, we construct a regulatory network model at the gene-program level to reveal the global architecture and logic of the regulatory network implied by transcriptional changes induced across a large number of gene knockdown experiments. Transcriptome-scale gene expression changes in response to gene knockdown perturbation reflect how cells coordinate the activities of major cellular functions to maintain homeostasis under the stress induced by gene knockdowns. To reveal the global architecture of perturbation responses, we focused on 3,138 gene-knockdown perturbations that are each associated with more than 10 deferentially expressed genes (DEGs)(STAR Methods 2.1). We coarse-grained the transcriptional profile into 30 gene programs, a number informed by both the Bayesian information criterion (BIC) and the elbow method, to balance the model's representative power and complexity [74] (STAR Methods 1.10). With 30 gene programs, 93.2% of the selected highly variable genes have a correlation larger than 0.5, with their corresponding gene programs in the gene matrix denoised by data diffusion (Figure S3F) [75].

The extracted gene programs reflect both general cell biology and lineage-specific gene programs that are specific to the biology of K562 cells (Figure 3B, Figure S3A, SI Table 3). We annotate the gene programs with a combination of bioinformatic databases including DAVID, Enrichr, and manual lookup [65, 66, 67]. Through oNMF, we extracted a series of programs that are enriched for genes involved in specific core cell processes including transcription, translation, DNA replication, mitosis, and RNA processing. D-SPIN also identified lineage-specific programs including an erythroid-specific program with hemoglobin (HBG1, HBG2, HGZ) and glycophorin (GYPA, GYPB, GYPE) genes, as well as two myeloid-associated programs with phagosome/actin-organization (ACTB, ACTG1, ARPC3) and immune-response (LAPTM5, VASP, RAC2) genes, respectively, which agrees with the multi-lineage potential of K562 cells (Figure 3B).

Given the set of 30 gene programs, D-SPIN generated a gene regulatory network model that provides a wiring diagram of K562 internal regulation and the response of the K562 regulatory network to gene knockdown perturbations (Figure 3B, Figure S3B, SI Data). The network construction took less than 8 hours with parallelization on 500 CPU cores. The D-SPIN network contains gene programs as nodes, and interactions between gene programs as edges that represent inferred interactions. The inferred network has low edge uncertainty quantified by Fisher information, as edge uncertainties are generally below 0.02 while strong interaction strengths are above 1. More than 98% of edges remain structurally stable when resampled from the posterior distribution of networks that represent near-optimal solutions to the network inference problem (Figure S3G, STAR Methods 1.6). D-SPIN identified 84 positive and 21 negative interactions between gene programs, corresponding respectively to 19.3% and 4.8% of all gene-program pairs. The D-

SPIN regulatory network reconstruction contains a set of sub-networks or modules of gene programs that interact with one another through dense positive (blue) interactions (Figures 3B and 3C). We automatically decomposed the D-SPIN network into seven modules through the Leiden community detection algorithm, and the module partition is robust to the hyperparameter choice [76] (STAR Methods 2.11). Each identified module reflects a core cellular function, including transcription, translation, protein-degradation, and cell-cycle-related functions.

The network contains a set of negative interactions between gene programs that are expressed in distinct cellular states. The strongest negative interaction is between gene programs expressed at different stages of mitosis. For example, the P29 Spindle microtubule has negative interactions with both P25 DNA replication and P27 Histone. The strong interactions inside the cell cycle sub-network of P25-P30 are able to reconstruct the transcription state distribution during cell cycle progression (Figures S3C and S3D). We also find strong negative interactions between distinct stress-response programs, P23 Unfolded protein response, and P2 Lysosome. The Unfolded protein response program consists of heat-shock protein (HSP90AB1, HSPD1, HSP90AA1) and other molecular chaperones (CCT8, CCT3, CCT5 genes), while the Lysosome program contains genes involved in lysosome genesis (PSAP, ASAH1, PLD3). We also observe negative interactions between P4 Erythroid and P6 Phagosome, where the phagosome is an innate immune response mounted by macrophages and other myeloid cell types [77]. The negative interaction between the Erythroid and Phagosome programs is consistent with the presence of two mutually exclusive differentiation paths that lead to erythroid and myeloid cell fates.

## D-SPIN reconstructs the architecture of core cellular pathways

In addition to inferring the structure of the core K562 gene regulatory network, D-SPIN also inferred interactions between each gene program and each gene knockdown perturbation. The interactions between perturbations and gene programs are represented by perturbation response vectors $\boldsymbol{h}^{(n)}$. The entries of $\boldsymbol{h}^{(n)}$ represent positive or negative interactions between perturbations and individual gene programs. The response vectors can be represented as directed edges that connect perturbations to the core regulatory network (Figure 3B).

Similarly to the grouping of the K562 gene regulatory network into interacting gene modules, we grouped gene knockdown perturbations into a set of 40 perturbation classes that we refer to as G1-G40 (Guide RNA group 1 through 40) through unsupervised Leiden clustering of the $\boldsymbol{h}^{(n)}$ vectors (STAR Methods 2.11, SI Table 3). Gene knockdowns within the same guide cluster have similar perturbation responses, suggesting that these genes are involved in the same pathway or have potential interactions. Identified clusters reflect the pathway-level organization in the K562 cell, revealing both well-known cell biology and more novel or cryptic organization of pathways (Figures 3D and 3E). Individual guide clusters are enriched for specific cell-biological functions, including DNA replication, the MAPK signaling pathway, and RNA degradation (Figures 3D and 3E).

The guide clusters also provide detailed information regarding the more granular structure of cellular pathways. For example, the components of the proteasome core particle (20S proteasome) fall into a single perturbation cluster (G3). In addition to these proteasome components, group G3 also contains genes involved in protein transport, including AKIRIN2 and IPO9 (Figure 3D), two protein components recently associated with the import of the proteasome into the cell nucleus through biochemical and genetic assays. In fact, in recent work, AKIRIN2 was found to directly bind to fully assembled 20S proteasomes to mediate their nuclear import with the nuclear import factor IPO9 [68]. This finding is consistent with the grouping of AKIRIN2 and IPO9 with proteasome components in the D-SPIN perturbation response clusters. Similarly, the D-SPIN guide clusters group the gene C7orf26 with the integrator subunits INTS10, INTS13 and INTS14, a key result of the original Perturb-seq study [28].

Thus, analysis of the perturbation response vectors inferred by D-SPIN provides a strategy for generating hypotheses regarding the architecture of core cellular pathways directly from single-cell mRNA-seq data. D-SPIN-generated hypotheses can, then, motivate biochemical experimental validation. We highlight the example of the proteasome and AKIRIN2/IPO9, as well as C7orf26 and integrator, to demonstrate that D-SPIN response vectors are consistent with two regulatory associations that were only recently validated through biochemical experiments in the literature [68, 28].

## Generating hypotheses for macromolecular complex sub-functions

Macromolecular complexes like the ribosome, the proteasome, the spliceosome, and the integrator complex play a critical role in cell physiology and gene regulation [78]. Protein complexes can often exist in different configurations with different combinations of sub-units. Defining the functional role of individual proteins within a complex and how protein components work together to achieve function is an important question in cell biology. Further, detailed insight into the architecture of specific proteins within protein complexes can identify therapeutic targets and has been a major focus in studying protein complexes in diseases including HIV, SARS-CoV-2, and cancer [79, 80, 81, 82]. Single-cell mRNA-seq experiments can provide insight into the functional roles of proteins within macromolecular complexes to provide hypotheses that can be studied in detailed but technically challenging biochemical experiments, including affinity purification and mass spectrometry [83].

Therefore, we used D-SPIN to gain insight into sub-functions of complex components by comparing cosine similarity between the response vectors $\boldsymbol{h}^{(n)}$ for components of known complexes in the CORUM database [84]. When subunits of a complex interact to achieve a common function, disruption of individual subunits will disrupt that common function, leading to high cosine similarity between subunit response vectors that approaches 1. When complexes can achieve distinct functions through combinations of distinct protein components, the knockdown of these components will generate different responses and the average cosine similarity approaches 0, which is the random expectation. We compiled a list of 46 core cellular complexes from the CORUM database by filtering out complexes that are not present in the dataset and merging variants of the same complex (STAR Methods 2.14). For each complex in the list, we defined a coherence score by computing the average cosine similarity between response vectors, $\boldsymbol{h}^{(n)}$, for all pairs of complex subunits (STAR Methods 2.14).

Across the 46 complexes in the CORUM database, our analysis revealed a wide range of coherence scores across different complexes, ranging from a low coherence score of 0.01 for the large Drosha complex to a high coherence of 0.73 for the mitochondrial large ribosome (Figure 3F, Figure S3H). Notably, complexes that play critical roles in specific biological functions including translation and metabolism, such as the mitochondrial ribosome, ribosome, exosome, DNA repair, and respiratory chain, exhibited the highest coherence scores. In contrast, complexes with known regulatory roles, including the large Drosha complex, DGCR8 complex, ALL-1 supercomplex (involved in chromatin regulation), the ATAC complex (histone acetyltransferase), and spliceosome all had lower coherence scores below 0.25. The low-coherence complexes are involved in versatile regulatory roles in epigenetic regulation and microRNA and mRNA processing. D-SPIN provides a list of the candidate cellular sub-processes (gene programs) that are controlled by individual sub-units of each macromolecular regulatory complex (SI Data), providing a specific framework for generating experimental hypotheses to determine how modulation of the subunit composition of these macromolecular regulatory complexes might enable the complex to achieve a range of regulatory functions.

As an example, the ribosome behaves as a coherent complex; the knockdown of individual genes in the large or small ribosome subunit has a similar impact on D-SPIN gene programs, as illustrated by the monochromatic edge connections between subunit perturbations and gene programs in the interaction diagram (Figure 3G). Perturbations to ribosome subunits activate protein degradation-related gene programs including P2 Lysosome, P3 Endosome, and P20 P21 Proteasome. The perturbations inhibit ribosome programs, P1 Mitochondria, P19 Endoplasmic reticulum, P23 Unfolded protein response, and P24 Oxidation. Thus, perturbing any ribosome subunits induces the cell into a stress response state with upregulated protein degradation and inhibition of protein production and secretion processes.

Unlike the ribosome, the proteasome and integrator behave as low-coherence complexes, and have response vectors that split subunits into smaller sub-groups that have distinct inferred interactions with downstream gene expression programs. The integrator complex subunits are partitioned into 4 sub-groups that agree with the original Perturb-seq study (Figure 3G): an INTS10-13-14-C7orf26 module, a cleavage module (INTS3, INTS4, INTS8, INTS11), and two shoulder and backbone modules (INTS2, INTS5, INTS7, INTS8; INTS1, INTS6, INTS12). Knockdown of the cleavage module activates P21 Proteasome and inhibits P22 rRNA processing, P18 Nuclear transport, P24 Oxidation, and P5 Lipid metabolism, while knockdown of the shoulder and backbone modules activates P4 Erythroid and P7 Innate immunity and inhibits P21 Proteasome, P28 Chromosome segregation, P17 RNA splicing, and P13 Ribosome programs.

Similarly, the proteasome complex subunits are split into two groups of core particles (20S proteasome) (PSMA6, PSMB1, PSMB3, PSMB6; PSMA2, PSMA7, PSMB2, PSMB7) and two regulatory particles (PSMC3, PSMC3, PSMC3, PSMD2, PSMD6; PSMD4, PSMD13). D-SPIN finds that the knockdown of pro-

teasome subunit PSMC6 (also known as TBP7 and Rpt4) has a negative impact on respiration; this result is consistent with the finding that PSMC6 interacts with mitochondrial proteins including TRAP1 in the endoplasmic reticulum to assist the folding of mitochondrial proteins [85]. We also identified signatures of protein complex sub-modules in the large Drosha complex, ALL-1 histone methyltransferase complex, and splicesome E complex. This indicates that specific subgroups of these complexes have different regulatory roles in controlling downstream gene expression programs. For example, the response vectors of spliceosome E complex subunits split into 4 sub-modules. Example genes in the 4 sub-modules (M1-M4) include heterogeneous nuclear ribonucleoproteins (hnRNPs)(M1), small nuclear ribonucleoprotein particles (snRNPs)(M2), core splicing factors (M3), and serine/arginine-rich (SR) proteins (M4) (Figure S3I). Different sub-modules can induce opposite perturbation responses on the same gene program: M1 activates P1 Respiration while M4 inhibits P1; M1 and M4 activate P4 Erythroid while M2 inhibits P4. These results demonstrate that the response vectors reveal the impacts of genetic perturbations, and provide a compact signature for identifying functional partitioning of molecular complexes.

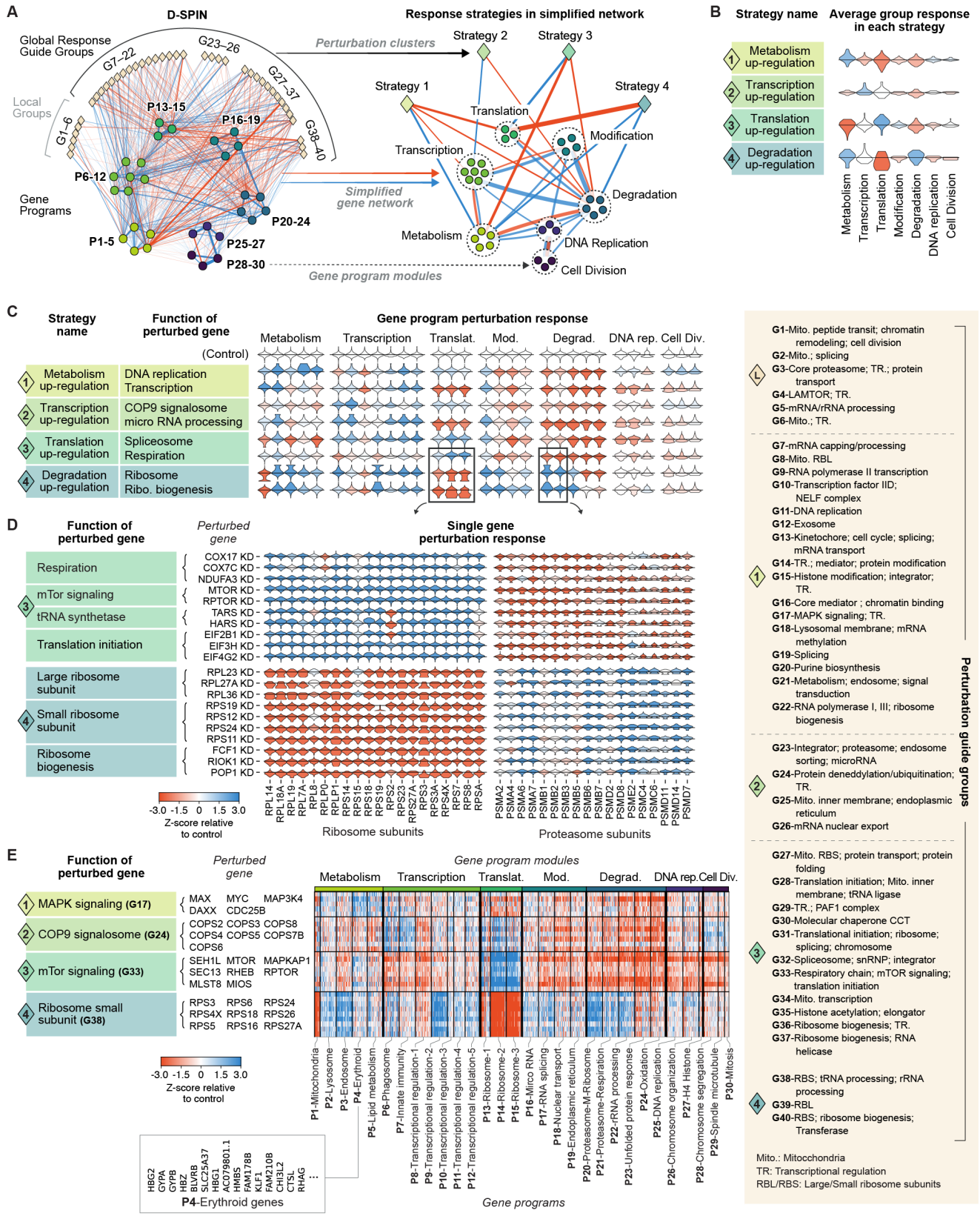


Figure 4: Caption on the following page

Figure 4 **D-SPIN network model identifies global perturbation response strategies in K562 cells for distinct classes of gene knockdowns.** (A) Diagram of network coarse-graining by grouping gene programs into 7 identified gene program modules and grouping guide RNA classes into 4 identified strategies. The resulting coarse-grained model enables global analysis of cellular regulatory responses. (B) Violin plots of averaged perturbation response vectors on programs in guide groups in each response strategy. K562 cells contain 4 distinct classes of global response strategies, and we name each strategy by the upregulated characteristic biological function. (C) Violin plots show gene program expression under knockdown of example pathway components relative to the control samples where a non-targeting guide RNA is introduced to cells. The upregulated biological function typically is distinct from the genes being perturbed. For example, when RNA polymerase is knocked down, cells generate the metabolism upregulation response, where the cell upregulates metabolism and downregulates transcription, translation, and degradation. As a second example, when ribosome subunits are knocked down, cells upregulate protein degradation and metabolism while downregulating translation. (D) Violin plots show the single-cell mRNA expression distribution for ribosome and proteasome subunits under knockdown of example genes in translation upregulation and degradation upregulation strategies relative to control samples. The gene-program level response strategies are reflected at the single-gene level. For example, when respiration-related gene COX17 or COX7C are knocked down, all ribosome subunit genes are upregulated and proteasome subunit genes are downregulated. (E) Gene-expression heatmap showing the upregulation (blue) or downregulation (red) of representative genes in all gene programs to the knockdown of individual genes in a series of cellular pathways relative to control samples. The program-level response strategies are shown as coherent upregulation/downregulation patterns of single genes in gene programs.

## Coarse-grained D-SPIN models provide insight into global perturbation response strategies

All cells must regulate a large number of internal processes (e.g., transcription, translation, cell cycle, metabolism, and differentiation) to maintain homeostasis in response to damage and perturbation. Understanding how cells control their physiological state is essential for an integrated understanding of cell regulation, but it remains poorly understood due to the lack of global integrated models. The D-SPIN gene-program model (Figure 3B) allows us to analyze the distributed regulation of core cellular processes in response to gene knockdown perturbations, a form of cellular damage.

While the D-SPIN model is compact, we find that the model can be simplified further through an automated coarse-graining strategy based on graph clustering (Figure 4A) that reduces the D-SPIN network by grouping gene programs into sets that we call *modules* and single gene perturbations into perturbation clusters. We call this strategy *network coarse-graining* because of its similarity to coarse-graining strategies in physical models, which group strongly correlated degrees of freedom into a single variable to aid computation and interpretation. The coarse-graining strategy produces a minimal network that we analyze to reveal 4 classes of stress-response strategies, patterns of gene activation, and inhibition generated by the cell when specific cellular processes are disrupted. The 4 strategies are consistent with the hyperparameter choice of the graph clustering (STAR Methods 2.12).

The D-SPIN-identified regulatory strategies suggest non-trivial forms of homeostatic regulation. We find that each strategy features the upregulation of a major biological function and downregulation of other functions; the only exception is that in the fourth strategy, both metabolism and degradation are upregulated (Figure 4B). The upregulation indicates that the cell compensates for the loss of a gene by upregulating a potentially compensating cellular function. The upregulation suggests the existence of active regulatory feedback within the regulatory network under the applied perturbations to maintain homeostasis. Therefore, the four global response strategies are named by the characteristic program upregulation as metabolism upregulation, transcription upregulation, translation upregulation, and degradation upregulation. In general, the K562 cells may be capable of mounting a wide range of homeostatic response strategies, but D-SPIN identified the four highlighted strategies given the specific experimental conditions in which the Perturb-seq experiment was performed.

For example, in the "translation upregulation" response strategy, perturbation to components impacting translation initiation (G28) leads to upregulation of translation and downregulation of protein degradation, suggesting that the cell compensates for decreased translational flux through a regulatory response that

both upregulates translation and downregulates processes that degrade proteins. Interestingly, perturbations to ribosome subunits and ribosomal rRNA (G38, G39, G40) production lead to a distinct response where the cell downregulates translation while upregulating protein degradation, suggesting a cellular attempt to contain the impact of unassembled ribosome subunits and damaged proteins (Figure 4C). The program-level perturbation responses are confirmed at the single-gene level. Under the knockdown of translation initiation factors (EIF2B1, EIF3H, EIF4G2), all ribosome subunits are coherently upregulated and proteasome subunits are coherently downregulated, as quantified by the Z-score relative to the control distribution of non-targeting guide RNAs (Figure 4D). Similarly, the knockdown of single large/small ribosome subunits or ribosome genesis genes (FCF1, RIOK1, POP1) induces coherent downregulation of all ribosome subunits and upregulation of proteasome subunits.

The regulatory strategies also provide insight into the flow of information in the cell. The compensatory functions are typically not directly associated with the gene being knocked down, suggesting long-range regulation or coupling between seemingly distinct processes. For example, a large proportion of metabolism upregulation responses are induced by perturbations to transcription processes, including RNA polymerase (G9, G22), Transcription Factor IID, NELF complex (G10), and mediator (G16). The knockdown of these genes leads to upregulation of metabolism, and downregulation of transcription, translation, and degradation. It has been found that transcription stress would induce an elevated ATP pool and rewired metabolism states [86]. As another example, the COP9 signalosome (CSN) is a protein complex involved in the deubiquitination and deneddylation of proteins. Knockdown of components including COPS2, COPS4, and COPS6 leads to upregulation of the phagosome and downregulation of protein-degradation machinery (Figure 4E). As another example, the knockdown of mTOR signaling components (which control cell growth) leads to the upregulation of ribosomal programs in a translation upregulation response.

These results demonstrate that D-SPIN-generated regulatory networks provide insights into global control strategies used by cells to regulate a distributed set of processes to achieve homeostasis in response to various forms of internal damage. Knockdown of translation initiation factors and ribosome subunits both affect the translation process, while the cells employ two distinct strategies (translation upregulation vs degradation upregulation) to maintain homeostasis, indicating the presence of a cellular sensing and information processing mechanism to distinguish and respond to different causes of stress. The correspondence between gene function damage and compensatory functions may lead to new therapeutic strategies, such as combinatorial inhibition of both a gene function and its associated compensatory pathways, for example, for improved killing of cancer cells. Such combinatorial strategies can be difficult to design due to the exponentially large space of potential combinatorial genetic interventions. Conceptually, the global perspective of D-SPIN illuminates classes of regulatory strategies, suggesting that compensatory mechanisms and information flow between cellular processes that might otherwise not be connected, thereby providing insights into principles of information processing and homeostatic control.

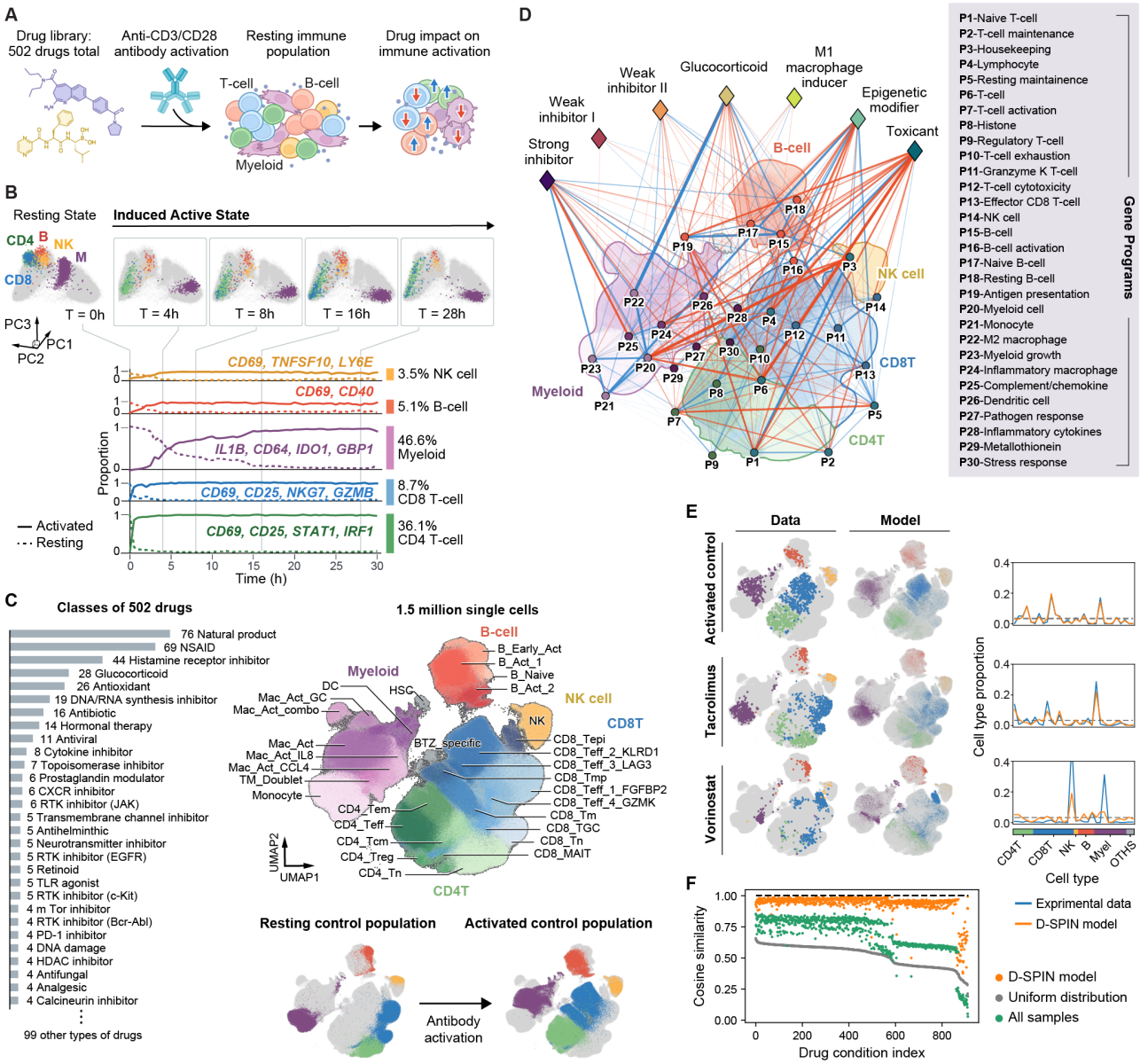


Figure 5: Caption on the following page

Figure 5 **D-SPIN derives a drug-response network model from human immunomodulatory drug-response single-cell mRNA-seq profiling experiments.** (A) Schematic of the experiment design for profiling drug responses on T-cell-mediated immune activation. Peripheral blood mononuclear cells (PBMCs) were harvested from a human donor. The cell population was treated by anti-CD3/CD28 antibodies that specifically activate T-cells and immunomodulatory drugs drawn from a library. The cell population was profiled after 24 hours of drug and antibody treatment. (B) (top) PCA projections derived from a 30-hour time-course experiment of T-cell-mediated immune activation, where samples were taken every 30 minutes for single-cell mRNA-seq. (bottom) The plot shows the time courses of the proportion of activated and resting cells in each cell type with example activation gene markers. T-cell reaches activated states first in 2 hrs and myeloid cell activation lasts 16 hrs. (C) (left) Histogram of classes of profiled drugs shows the selected drugs have a variety of biochemical properties and target pathways. (right-top) UMAP embedding of 1.5 million filtered single cells obtained from the drug profiling experiments [87]. In the profiled cell population we identified 32 cell states in the major cell types of T-, B-, NK, and myeloid cells, and each cell state is curated by marker genes and gene differential expression. (right-bottom) UMAP embedding of the resting control cell population (without antibody activation) and activated cell population. The resting and activated cell states compose the major partition on the UMAP. (D) Network diagram shows the inferred regulatory network model between 30 gene programs (P1-P30, circles), as well as interactions between programs and 7 drug classes (diamonds) identified by D-SPIN through clustering the response vectors. In the network rendering, each program is positioned on the UMAP schematic upon the cell state where the program is highly expressed. (right box) Gene programs are functionally annotated through gene ontology annotation tools [65, 66, 67] and manual lookup. (E) (left) UMAP embeddings of experimental cell state distribution compared with the state distribution generated by the D-SPIN model in example conditions including activated control, Tacrolimus treatment, and Vorinostat treatment. (right) Line plots quantify the distribution of cell states of experimental data and D-SPIN models in each example condition. The dashed line in cell state distribution is uniform distribution as reference. D-SPIN models agree well with experimental data in the control and Tacrolimus-treated samples. The model fits less well in the Vorinostat-treated sample but still captures qualitative features of the distribution. (F) Scatter plots show cosine similarity between experimental data and cell-state distributions generated by D-SPIN, compared with two reference null models. One null model is a uniform distribution over cell states, and the other null model is pooling all cells from samples of all experimental conditions to reflect the relative abundance of cell states. The cosine similarities between cell state distributions of D-SPIN models and experimental data are higher than 0.9 for 92.4% of conditions.

## Modeling immunomodulatory drug responses in primary human immune cells

D-SPIN models can contribute to the design of therapeutic strategies by providing global insights into how different underlying cell types respond to a given small molecule or other therapeutic intervention. Given the role of the human immune system in many diseases, we applied D-SPIN to dissect the response of human immune cells to a library of small-molecule immunomodulatory drugs. We developed an experimental dataset in which we applied single-cell mRNA-seq to profile the response of human immune cells to 502 different immunomodulatory drugs. Our experimental system was designed to probe the impact of drug modulation on a cell population undergoing T-cell driven hyper-activation, as is observed in auto-immune and hyperinflammatory states [88, 89]. We cultured a heterogeneous population of primary donor-derived human immune cells (peripheral blood mononucleated cells, PBMC) that contain T-cells, B-cells, myeloid cells, and NK-cells. T-cells were activated using T-cell receptor antibodies anti-CD3/CD28 (Figure 5A, Figure S4A, STAR Methods 1.12), leading to the activation of macrophage, B-cell, and NK-cell cell types (Figure 5B, Figures S4B-S4D, STAR Methods 1.13). We applied the system to analyze the action of different immunomodulatory drugs on the activation of the immune cell population.

Single-cell mRNA-seq was used to measure the effect of the 502 immunomodulatory small molecules on T-cell-driven-immune activation (Figure 5C). The drug library contains a diverse set of small molecules targeting pathways including mTOR, MAPK, glucocorticoids, JAK/STAT, and histone deacetylases (HDAC) (Figure 5C, SI Data). The library contains drugs used for treating auto-immune disease (e.g., Tacrolimus, Budesonide, Tofacitinib) as well as FDA-approved anti-cancer drugs (e.g., Bosutinib, Crizotinib). We added each compound at 1μM at the same time as the anti-CD3/CD28 antibody, and profiled the cell population using single-cell mRNA-seq at 24 hours [29, 90]. For additional context, we also profiled drug responses in resting

cell populations without CD3/CD28 activation for a subset of drugs. In total, we profiled 1.5 million single cells in resting and activated conditions, with over 1,200 total conditions and 31 different immune cell states, including 4 CD4 T-cell states, 10 CD8 T-cell states, an NK-cell state, 4 B-cell states, and 8 myeloid cell states (Figure 5C, Figure S4E, STAR Methods 2.2, SI Table 5).

Using the single-cell mRNA-seq data, we applied D-SPIN to construct a regulatory network model of the cell population and the interactions of individual small molecules with the network to drive the observed drug impacts. The heterogeneous population contains various immune cell types as well as resting, activated, and drug-specific states. We applied D-SPIN in an unsupervised manner, using the algorithm to construct a single, integrated regulatory network model that can capture all cell types and cell states by pairwise interactions between gene-expression programs and interactions between gene programs and applied drugs.

For network construction, we chose 30 gene programs as informed by both the Bayesian information criterion (BIC) and the elbow method on gene information capturing [74] (Figure S5D, STAR Methods 1.10). With 30 gene programs, 89.0% of the selected highly variable genes have correlations larger than 0.5 with corresponding gene programs in the gene matrix denoised by data diffusion [75]. We performed oNMF with seeding to include drug-specific programs, and the programs were sufficient to describe the transcriptional state of all cells in the population (Figures S5A and S5B, STAR Methods 2.3, SI Table 5). The decomposition yielded known T-cell and macrophage gene-expression programs, including T-cell resting (P5) and activation programs (P7), macrophage anti-inflammatory (P22 M2 macrophage), pathogen response (P27), antigen presentation (P19), and global cell-state programs for T-cells (P6), B-cells (P15), NK cells (P14), and myeloid cells (P20). The function of each program was annotated by a combination of informatics databases [65, 66, 67] and manual annotation, and subsequently validated on a single-cell atlas of human immune cells [91].

Using the experimental data and the gene expression programs, D-SPIN inferred a single interaction network $\boldsymbol{J}$ and a set of drug-specific response vectors $\boldsymbol{h}^{(n)}$ across all treatment conditions (Figure 5D, SI Data). The inferred network has low uncertainty, as edge uncertainties are generally below 0.06, while strong interactions have strength above 1. Further, more than 97% of edges remain structurally stable when resampled from the posterior distribution of near-optimal networks (Figure S5E, STAR Methods 1.6). Like the K562 cell network, the immunomodulatory drug-response network has a distinct modular structure, with a set of core network modules composed of tightly interacting gene programs (Figure 5D, Figure S5C). The inside-module interactions are mostly positive and account for 76.2% of strong activating edges in the network with strength larger than 1. The modules in the drug data contain groups of gene programs that are expressed together in individual cell types. For example, a module corresponding to anti-inflammatory myeloid cells contains P21 Monocyte, P22 M2 macrophage, and P23 Myeloid growth and a network hub P20 Myeloid cell. Similarly, in T-cells, another module containing P11 Granzyme K T-cell, P12 T-cell cytotoxicity, P13 Effector CD8 T-cell, and P14 NK cell corresponds to the population of CD8 T cells and NK cells. In the D-SPIN model, negative interactions primarily occur between gene programs expressed in different cell types, e.g., P2 T-cell maintenance and P19 Antigen presentation, as antigen presentation happens in B-cells and myeloid cells but not T-cells (Figure 5D).

As a maximum entropy model with a minimal number of parameters, the D-SPIN model only contains a total of 465 network interaction parameters and 30 parameters for each drug condition. However, it can generate samples that are consistent with all profiling experiments of 502 drugs (STAR Methods 2.13). Qualitatively, we found visual similarity between the distribution generated by the D-SPIN model on the UMAP embedding and cell state distributions (Figure 5E). Quantitatively, we computed the cosine similarity between the model and data distribution on cell states, and in 92.4% of the samples the model is $> 90\%$ similar to data (Figure 5F). The cell state distributions produced by D-SPIN also exhibit significantly higher similarity with data distributions compared to the reference distribution obtained by pooling all experimental conditions together to reflect the relative abundance of cell types ("all samples" in Figure 5F). The drugs with lower cosine similarity tended to be drugs that drove the cell population into a few highly specialized cell states, like the proteasome-inhibitor Bortezomib, which produced a strong signature of cell stress in all cells. For these drugs, the D-SPIN model produced a cell-population distribution that had a qualitative agreement with the data but differed quantitatively, likely due to the extremely focused cell-type distribution for these drugs. To further access the distributions generated by the D-SPIN model, we employed several other metrics for comparing high-dimensional distributions. Specifically, the errors between data and model for the sample mean and cross-correlation by D-SPIN are, respectively, 3.9% and 7.2% compared to the uniform distribution over all possible states, and the optimal transport distance by D-SPIN distributions is 12.4% of the uniform distribution as a distribution-wise metric (Figure S5F). At the single-state level, the probabilities of states

in D-SPIN models correlate with the empirical probabilities of states in experimental data with a correlation of 0.54, 0.58, and 0.48 in example conditions of activated control, Tacrolimus treatment and Vorinostat treatment; the probabilities are not in exact numerical agreement due to the entire high-dimensional space of $3^{30} \approx 10^{14}$ states being far under-sampled by the thousands of cells in each condition in the experimental data (Figure S5G, STAR Methods 2.13).

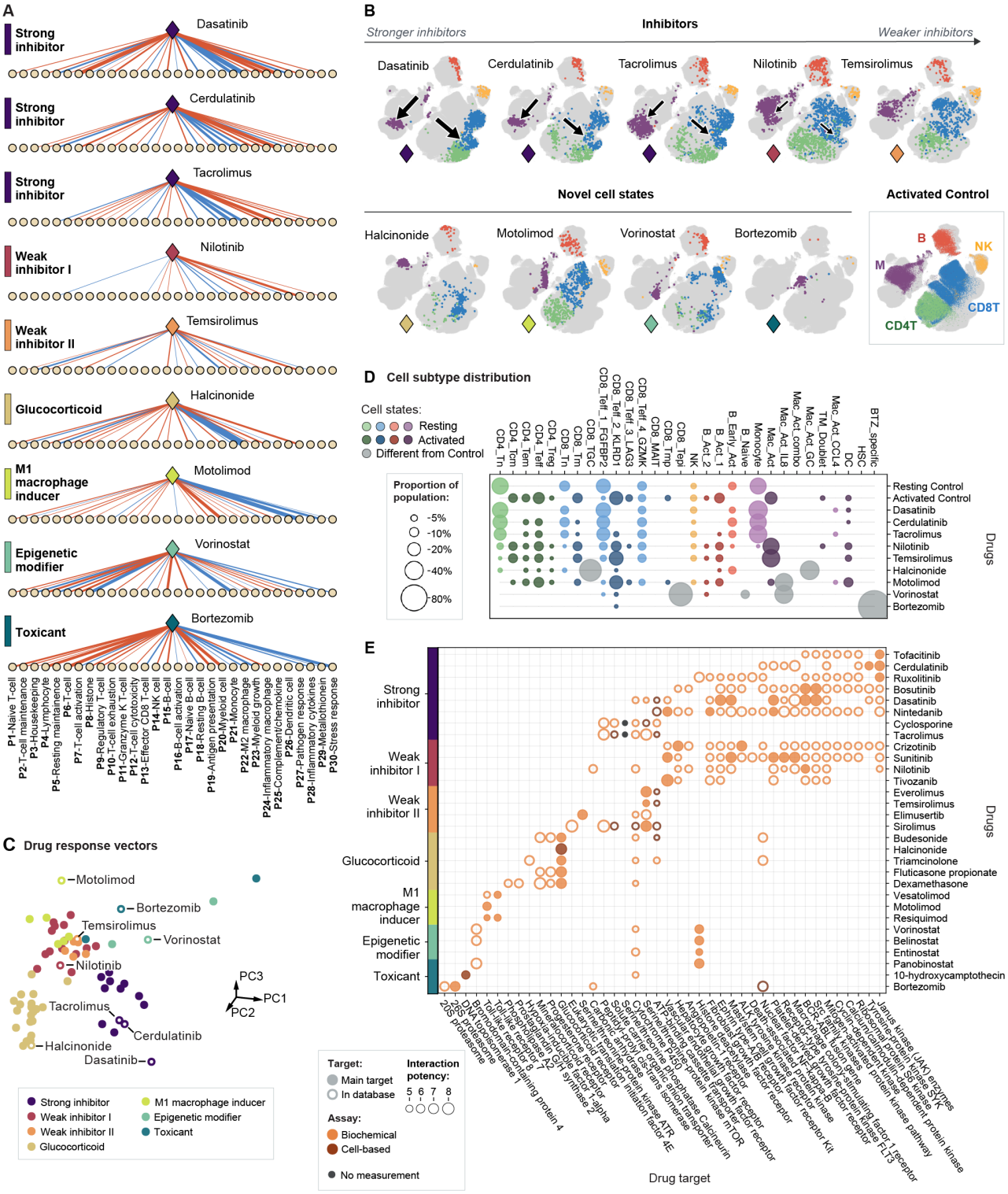


Figure 6: Caption on the following page

Figure 6 **Drug classification derived from D-SPIN aligns with known drug targets and mechanisms of action.** (A) Diagrams of response vectors inferred by D-SPIN for example drugs show positive/negative interactions between the drug and gene-expression programs as blue/red edges. Edge thickness scales with interaction strength. Gene program names for all diagrams are indicated at the bottom of the plot. D-SPIN identifies 7 phenotypical classes of drugs based on their response vectors. (B) UMAP embeddings of immune cell population after treatment of antibody and example drugs from different classes. The top row shows 5 immune inhibitors with decreasing strength from left to right, from complete suppression of immune activation (Dasatinib) to slightly increased resting cell population (Temsirolimus). The bottom row shows 4 example drugs that each induce novel cell states distinct from both the resting and activated control cell population. Cell state distribution of the activated control population (bottom-right) is shown for comparison. Using response vectors, D-SPIN identifies immune inhibitors with a spectrum of strengths and drugs that produce novel cell states. (C) Scatter plot visualizes the response vectors of identified effective drugs that are statistically different from control in the space of leading principal components (PCs). Each drug is colored by its drug class. (D) Bubble plots show cell-subtype distributions induced by selected drugs as well as the control cell population. Bubbles have sizes scaling with the proportion in the population, and are colored by major cell types and resting/activated classification. With decreasing inhibitor strength, the proportion of the activated immune-cell population (deep colors) gradually increases. Some drugs induce cell states that are different from both resting- and activated-control populations. (E) Bubble plots display the biochemical targets, primary targets, and affinities of example drugs obtained from database [92, 93]. Bubbles have sizes scaling with interaction potency quantified by biochemical (orange) or cell-based (red) assays. Primary targets are represented in solid dots and other targets in open dots. Drugs in the same phenotypical class identified by D-SPIN can act on a variety of molecular targets and pathways.

## Classification of small molecules using D-SPIN

In addition to the regulatory network model, D-SPIN also inferred a perturbation response vector $\boldsymbol{h}^{(n)}$ that quantifies interactions between each drug and the regulatory network. By comparing the inferred response vector with control experiments, we identified 70 out of 502 drugs that had statistically significant interactions with the core gene network (STAR Methods 2.11). Moreover, as in the case of the Perturb-seq data, we applied clustering to group drugs into a set of 7 classes, including strong inhibitor, weak inhibitor I, weak inhibitor II, glucocorticoid, M1 macrophage inducer, epigenetic regulator, and toxicant, based on their interactions with the underlying regulatory network (Figure 5D, Figures 6A, SI Table 6). We named the classes by analyzing the structure of the cell population induced by drugs within the class. In addition to inducing similar transcriptional responses in the underlying cell population, the drug classes identified by clustering of D-SPIN perturbation response vectors, in many cases, belong to specific chemical families with common mechanisms of action.

Three of the seven drug classes are drugs that inhibit activation of the immune-cell population (Figure 6A, Figure S6A), leaving T-, B-, and myeloid cells in the resting state, including drugs we classified as strong inhibitors, weak inhibitor I, and weak inhibitor II. Across these drug classes, analysis of cell state distributions and UMAP visualizations indicate that these drugs block the transition of T-, B-, and myeloid cell types to the activated state, shifting the balance of the cell population to resting cell states, but acting with a spectrum of inhibition strengths (Figures 6B-6D). Very strong inhibitors, including the immunosuppressive drug Tacrolimus and the cancer drug Dasatinib, completely block the activation of T-cells and myeloid cells, inducing a cell population with a similar cell-state profile to unstimulated PBMC population (Figure 6D). Weak inhibitors, such as Temsirolimus, only slightly bias the cell population towards resting states, compared with the activated control population.

The individual drugs that fall into the inhibitor drug classes act on a variety of molecular targets and signaling pathways [92, 93] (Figure 6E, SI Data). The strong-inhibitor class targets include JAK (Tofacitinib, Cerdulatinib, Ruxolitinib), BCR-ABL and Src family kinases (Bosutinib, Dasatinib), and calcineurin (Cyclosporine, Tacrolimus), which are all pathways downstream of T-cell receptor signaling. The drugs in the weak inhibitor I class target another group of receptors including ALK (Crizotinib), FLT3 (Sunitinib), and various growth-factor receptors (Crizotinib, Sunitinib, Nilotinib, Tivozanib). The drugs in the weak inhibitor II class target protein kinases including mTOR (Everolimus, Temsirolimus, Sirolimus) and ATR (Elimusertib). Broadly, the analysis suggests that inhibition can be achieved via a range of distinct biochemi-

cal pathways and mechanisms. However, the strongest inhibitors in our data have a high affinity for signaling molecules immediately downstream of T-cell receptor activation, including the Src family kinase Lck.

While many of the drugs in our dataset have been previously reported as immunosuppressive, including well-known drugs such as Tacrolimus, our single-cell profiling measurements and D-SPIN analysis organized the drugs into a quantitative series and provided a single integrated framework for comparing drug action at the transcriptome scale across a large number of inhibitors with different chemical mechanisms of action.Specifically, immunomodulatory drugs vary in the breadth and specificity with which they interact with biochemical targets, and a therapeutic question is to understand how differences in biochemical preferences translate into transcriptional changes and thus differences in therapeutic response [94, 41, 95]. Our dataset provides the opportunity to perform a detailed comparison between the effects of drugs with the same primary target but different breadth and specificity of secondary targets. Thus, the single-cell assay and D-SPIN model together can identify phenotypic patterns that capture potential differences in the outcome of drug action (Figure 6E).

To quantify and rank the strength of immune inhibition by drugs, we computed an inhibitor strength for each immune inhibitor drug by projecting the D-SPIN drug response vector onto the leading singular vector of all inhibitors in our dataset (Figure S6C, STAR Methods 2.16). The resulting quantification of inhibitor strength, when combined with information on biochemical targets (Figure 6E), revealed potential principles of immunosuppression. For example, a specific set of BCR-ABL kinase inhibitors (Dasatinib, Bosutinib) developed for the treatment of leukemia generated strong inhibition with strengths of 6.57 and 4.61, while another set of BCR-ABL inhibitors, including Nilotinib, fall into the weak-inhibitor class with a strength of 1.67. Biochemically, the BCR-ABL strong inhibitors can be distinguished from the BCR-ABL weak inhibitors by their increased Lck/Src kinase specificity (Figure 6E). The Lck kinase is a central signaling node directly downstream of T-cell receptor activation. Similarly, both as strong inhibitors, the Janus Kinase (JAK) inhibitor Cerdulatinib has a stronger inhibitory strength of 5.31 than the FDA-approved anti-inflammatory JAK inhibitor Tofacitinib with a strength of 3.75. While both drugs have JAK as their primary target, Cerdulatinib also targets the kinase SYK. Therefore, the response vectors generated by D-SPIN, when combined with data on drug-target binding (Figure 6E), allowed us to identify potential signaling nodes like Lck and SYK that differentiate the transcriptional responses of otherwise related drugs like the JAK inhibitors Cerdulatinib and Tofacitinib [94].

Beyond the strong and weak inhibitors, D-SPIN identifies a class of drugs that contains glucocorticoids (GCs), steroid-derived small molecules that activate the glucocorticoid receptor (Figures 6A-6D). The GC class of drugs includes well-known immunosuppressive drugs including Halcinonide, Budesonide, Triamcinolone, and Dexamethasone. The GCs generate a population response that is similar to the strong inhibitors like Dasatinib. However, the two classes of drugs also have important differences in their interaction with the D-SPIN regulatory network. Specifically, D-SPIN infers that the GCs not only inhibit activation programs in T-cells and myeloid cells, but also activate a specific immunosuppressive gene program in macrophages. The program is named P22 M2 macrophages, as it contains genes associated with the immunosuppressive M2 macrophage including CD163 [96, 97]. Therefore, D-SPIN infers important differences in the immunosuppressive effect of GCs and other inhibitors. GCs have weaker suppression of programs associated with chemokine secretion (P25), but induce the M2 macrophage program that is more weakly activated by drugs in the strong-inhibitor classes.

In addition to the suppressive drug classes, D-SPIN also identifies a group of drugs that induce the activation of inflammatory, pathogen-responsive M1 macrophages. The class includes small-molecule activators of Toll-like receptors, receptors in macrophages that sense the presence of pathogens and whose activation induces an innate immune response. The M1 macrophage inducer class contains TLR7 agonists (Vesatolimod, Resiquimod) and TLR8 agonists (Motolimod, Resiquimod), and produces macrophage states related to host defense that highly express P27 Pathogen response and P29 Metallothionein.

Finally, D-SPIN identifies two classes of drugs, which we call epigenetic modifiers and toxicants, that the model associates with activated cell-stress programs. Toxicants include the proteasome inhibitor Bortezomib, potent non-selective histone deacetylase inhibitor Panobinostat, and DNA topoisomerase inhibitor 10-Hydroxycamptothecin. Toxicants strongly activate the P30 Stress response and have mostly inhibitory interactions with other gene programs, especially generic cell-type programs such as P6 T cell, P20 Myeloid cell, and P4 Lymphocyte. The epigenetic-modifiers class consists of histone deacetylases (HDAC) inhibitors which generate an epigenetically disrupted T-cell state (CD8_Tepi) that has elevated expression of histone-related genes such as histone component genes and DNA topoisomerase TOP2A.

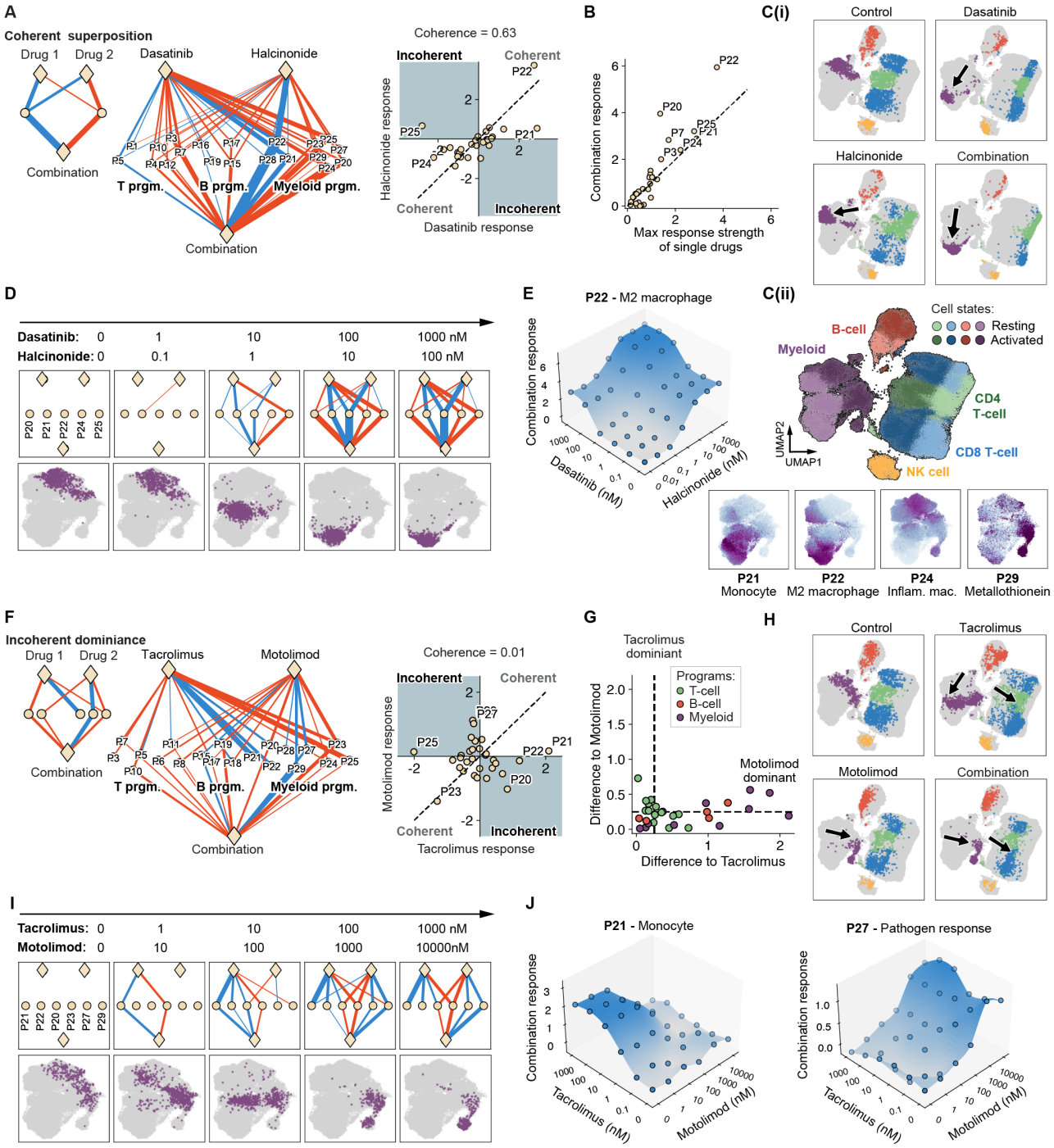


Figure 7: Caption on the following page

Figure 7 **D-SPIN reveals network-level mechanisms of combinatorial drug action.** (A) (left) Schematic diagram of coherent superposition where the single drugs both have activation/inhibition effects on downstream gene programs and the combination drug response is the superposition of single drug effects. (middle) Diagram of response vectors for Dasatinib and Halcinonide alone and their combination. The two single drugs and their combination have coherent effects on gene programs by activating/inhibiting the same set of programs. (right) Scatter plots compare the response vectors on each gene program for Dasatinib and Halcinonide. The two drugs have a high coherence score (0.63) quantified by the cosine similarity, but different strengths of interactions with gene programs. (B) Scatter plots of the response vectors on each gene program for the drug combination and max response strength of two single drugs. The combination responses on gene programs are a superposition of single-drug responses, as they are generally higher than the maximum strength of the single-drug responses. (C)(i) UMAP embeddings of activated control, single-drug-treated, and drug combination treated cell population, with arrows indicating different myeloid cell state changes. The drug combination induces a novel macrophage state that occupies a distinct region on the UMAP. (ii) (top) UMAP embedding of cell states of the two drug dosage titration experiments, Dasatinib plus Halcinonide and Tacrolimus plus Motolimod. Cell states are colored by major cell types and resting/activated classification. (bottom) UMAP rendering of example myeloid programs, where cells are colored by the expression level of each gene program. (D) (top) Diagram of response vectors for two single drugs and their combination on most impacted myeloid programs under different dosages. The combination response is a superposition of coherent single-drug interactions across dosages. (bottom) UMAP embeddings of myeloid states for each drug combination dosage. The novel macrophage state is induced by a combination of gene program recruitment of single drugs. (E) Surface plot shows response vectors on program P22 under different dosage combinations, where dots are from experimental data and the surface is spline interpolation. On the program, each single drug response exhibits a sigmoid-shaped curve with the logarithm concentration but with different saturating strengths, and the combination response is approximately the sum of the two sigmoid-shared single drug responses, showing the signature of superposition/additivity. (F) (left) Schematic diagram of incoherent dominance where the single drugs have different activation/inhibition effects on gene programs and the combination response is dominated by one of the single drugs. (middle) Diagram of response vectors for Tacrolimus and Motolimod alone and their combination. The two drugs have incoherent effects as they activate/inhibit different sets of gene programs, and the combination effects follow one of the single drugs. (right) Scatter plots compare the incoherent response vectors of the two drugs with a low coherence score (0.01). (G) Scatter plots of the difference between combination response and two single drug responses. On myeloid and B-cells, the response is more similar to, therefore dominated by Motolimod. On T-cells, some programs are dominated by Tacrolimus and some are dominated by Motolimod. (H) UMAP embeddings of activated control, single-drug-treated, and drug combination treated cell population, with arrows indicating different cell state changes. The drug combination induces a novel cell population state with inhibited T-cell states and Motolimod-induced-like myeloid states. (I) (top) Diagram of response vectors for two single drugs and their combination on most impacted myeloid programs under different dosages. At high concentrations of Motolimod, the activations of Tacrolimus on P21 and P22 are dominated by the no-effect of Motolimod. (bottom) UMAP embeddings of myeloid states for each drug combination dosage. The dominance of Motolimod over Tacrolimus on myeloid programs induces myeloid states similar to the effect of Motolimod alone. (J) Surface plot shows response vectors on (left) program P21 and (right) program P27 under different dosage combinations. On the program P21, the activation effect of Tacrolimus is abolished by increasing the dosage of Motolimod. On the program P27, the activation of Motolimod is not influenced by the dosage of Tacrolimus.

## Drug combinations that generate hyper-suppression

Immunomodulatory drugs are often used in combinations, and drug combinations could in principle be used to tune the transcriptional cell-state distribution of the immune cell population to meet therapeutic goals. However, the design of drug combinations at the transcriptome scale is challenging due to a large number of potential drug-gene interactions. D-SPIN models provide a framework to compare the action of individual drugs on gene expression programs within the regulatory network in order to identify drug combinations with potentially useful therapeutic applications.

Therefore, we selected 10 drugs from different drug classes identified by D-SPIN, profiled all pairwise

combinations experimentally, and applied D-SPIN to interpret the mechanisms of combination action (Figure S7A). By comparing the D-SPIN inferred response vectors of drug combination with single drugs, we found that most of the drug interactions are additive or subadditive on the gene program level where the effect of drug combination is equal to, or slightly weaker than, the linear sum of single drug effect. Other types of drug interaction include dominant (single drugs have different effects on the gene program and the combination follows one of them), synergistic (the combination is stronger than the linear sum of single drug effects), and antagonistic (the combination is weaker than each single drug effect). Quantitatively, 83.0% of drug interactions are additive or subadditive on gene programs, 9.4% are dominant, 6.2% are synergistic, and 1.4% are antagonistic (STAR Methods 2.17, SI Table 7). The additive interactions between drugs can recruit a combination of transcriptional programs from single drugs, creating novel cell states or population states, especially between drugs that have distinct impacts.

To determine how the additivity between single drugs on different programs generates novel biological outcomes, we first investigated the network-level similarities and differences between glucocorticoids (GCs) and strong inhibitors. In our experiments, GC action on the D-SPIN regulatory network was similar to the action of strong inhibitors, with both classes of drugs suppressing the activation of T-cells, macrophages, B-cells, and NK-cells. However, GCs induce weaker suppression, allowing cells to activate inflammatory cytokines, including CXCL10 and CXCL11 in the P25 compliment/chemokine program, whereas strong inhibitors, including the Src family inhibitor Dasatinib, strongly suppress these cytokine programs. GCs uniquely induce an M2 macrophage gene-expression program, which could be useful in some therapeutic settings, due to the expression of anti-inflammatory signaling molecules. The difference in impact between GCs and strong inhibitors is seen in the comparison of the inferred interactions with the network of the GC Halcinonide with the inhibitor Dasatinib (Figure 7B). Gene programs P22 M2 Macrophage, P25 Complement/chemokine, and P21 Monocyte are induced or repressed more strongly by one drug than the other.

Therefore, we investigated whether combinations of glucocorticoids and strong inhibitors could induce both the M2 macrophage activation and cytokine-repression effects, by performing single-cell mRNA-seq on cell populations treated with the glucocorticoid Halcinonide in combination with the strong inhibitor Dasatinib across a range of doses (Figure S7B). We used D-SPIN to analyze the impact of the drug combination on the regulatory network and underlying cell population. We found that the drug combination induces both the strong immune suppression of Dasatinib as well as the induction of M2 macrophage gene-expression programs for a specific range of concentrations (Figures 7B-7D). The response vectors for the drug combinations show that the impact of Dasatinib and Halcinonide are additive at the gene-program level across combinations of different concentrations. The response vectors for the drug combination indicate significantly higher P22 M2 macrophage induction than the single drug effect of either Dasatinib or Halcinonide (Figure 7E), while also exhibiting increased suppression of P24 Inflammatory macrophage over the single drugs. The response vectors also maintain strong activation of the resting program P21 Monocyte. However, the combinatorial responses do not exceed the sum of individual drug effects, so we classify the effect as additive or subadditive rather than synergistic. Similarly, on T-cell programs, the drug combination causes a subadditive activation on P5 Resting maintenance, and additive inhibition on P7 T-cell activation and P12 T-cell cytotoxicity (Figure 7A).

The combination of Dasatinib and Halcinonide, therefore, generated a hyper-repressed cell population with a repressed T-cell population and a macrophage cell population with activation of both resting myeloid program P21 and M2 macrophage program P22 (Figure 7C). The modulation is monotonic with doses as reflected on the response vectors, and manipulation of the drug doses allowed conversion of the cell population from activated to intermediate suppression (resting-like) to hyper-repressed M2 macrophage-like states (Figures 7D and 7E, Figure S7D, STAR Methods 2.19). Conceptually, by combining the two drugs, we are effectively able to fine-tune the transcriptional state of the macrophage population by exploiting a superposition interaction. Here, the drug combination recruits a combination of gene-expression programs in a dose-dependent fashion, which is also observed in a previous study performed at the single-protein level [98].

## Drug combinations that activate the innate immune system while repressing T-cell activation

In a variety of therapeutic settings, it can be advantageous to modulate the balance between adaptive (T-cell-driven) and innate (macrophage-driven) immune responses. For example, T-cell responses may react

adversely with transplanted organs and are therefore suppressed in transplant patients. Unfortunately, infections become a significant cause of mortality as a result of immunosuppression. Thus, the induction of an innate immune response in the background of immunosuppression provides a strategy for combating infection [99, 100, 101]. To understand whether D-SPIN could be applied to tune the balance between different immune cell states at the population level, we use D-SPIN model-identified M1 macrophage-inducing small molecule Motolimod, as well as standard immunosuppressive small molecules, including Tacrolimus, which are revealed to be mostly incoherent with Motolimod in their impact on the underlying gene regulatory network (Figure 7F). Specifically, the global cosine similarity between Motolimod and Tacrolimus responses was only 0.01 and the two drugs are biased to impact T-cell (Tacrolimus) or macrophage (Motolimod) gene-expression programs, with Tacrolimus activating resting T-cell programs and Motolimod activating a pathogen-responsive macrophage program, P27 (Figure 7G).

To exploit the incoherence between Tacrolimus and Motolimod, we added the drugs in combination across a series of doses, applied single-cell mRNA-seq to profile the resulting cell population, and analyzed the combinatorial drug response with D-SPIN (Figure S7C). D-SPIN identifies a concentration regime (Figure 7H, Figure S7E) in which the cell population contains activated macrophages with activated pathogen response program P27 while maintaining T-cell repression with inhibited T-cell activation program P5, with higher than 1000nM Motolimod and higher than 100nM Tacrolimus (Figure 7I). Specifically, the two single drugs have different impacts on myeloid programs: Tacrolimus activates the resting program P21 Monocyte, and Motolimod activates the pathogen response programs P27. The drug combination exhibits non-additive, dominant interactions on the programs P21, where the combination follows the no-effect of Motolimod instead of the activation of Tacrolimus (Figure 7J). The non-additive interactions are important because Motolimod's dominance over Tacrolimus on P21 leads to the pathogen-responding M1 macrophage state, similar to the state induced by Motolimod alone. Conceptually, the Motolimod and Tacrolimus drug combination demonstrates that drugs impacting separate cell types can be combined to tune the structure of an immune cell population.

# Discussion

Here, we introduce D-SPIN, a probabilistic modeling framework, that constructs quantitative, generative models for gene regulatory networks from large-scale single-cell mRNA-seq datasets collected from cell populations across a series of perturbation conditions. The D-SPIN framework is general and can be applied to study the response of cells to genetic perturbations, small molecules, and signaling conditions. D-SPIN models can be treated both as network models that are analyzed using the tools of graph theory as well as complete probabilistic models that can predict the structure of a cell population across a series of different conditions. The mathematical structure of D-SPIN allowed us to develop a computationally efficient, parallel inference procedure that can be run on hundreds of CPU cores to perform network inference on datasets with thousands of perturbations and millions of cells. Our work establishes an interpretable model framework for transcriptome-scale perturbation datasets. The D-SPIN framework is a maximum-entropy modeling framework adapted from the analysis of systems in condensed-matter physics [32]. D-SPIN provides a low-dimensional, compact yet predictive framework for analyzing population-level effects in single-cell data through a regulatory network model. Biologically, we apply D-SPIN to generate global insights into network-level mechanisms of cellular homeostasis (Figure 4), to define hypotheses regarding the sub-functions of protein complexes (Figure 3), to demonstrate the additive logic of combinatorial drug responses at the transcriptome scale, and thus to develop a framework for applying the additivity of immunomodulatory drugs to program immune cell-states and immune population structure (Figure 7).

Single-cell mRNA-seq methods enable large-scale perturbation response studies across cell types and organisms. D-SPIN provides a computational framework for integrating information from such studies into regulatory network models that can be analyzed and compared to reveal principles of cellular regulation. We have demonstrated that D-SPIN provides insight into damage response strategies in K562 cells. Stress response strategies play a role in many diseases including cancer, and diseases of ageing and degeneration. Further, cell response strategies change over the lifetime of organisms and also over evolutionary time scales. Our understanding of gene regulatory network architecture, logic, and evolution has been limited by the limited number of network reconstructions that exist. D-SPIN provides a framework to construct, analyze, and compare regulatory networks to understand how they change over time.

Cells are distributed control systems that modulate many internal processes to maintain homeostasis

and execute cell-fate transitions. However, the principles of distributed control at the transcriptome scale are poorly understood. D-SPIN models help reveal the principles of distributed control. We show that K562 cells mount a set of regulatory responses when different internal processes are perturbed through gene knockdown. The response strategies point to a broadly distributed control where long-range information flow between distinct cellular processes, such as coherent up/down regulation of proteasome/ribosome subunits under ribosome subunit knockdown, points to the presence of sensing and controlling mechanisms in the cell. Identifying these feedback control points with combinatorial knockdown screening may lead to the discovery of new regulators or regulating mechanisms.

D-SPIN can also assist the design of interventions that modulate networks and cell states. We showed that D-SPIN can be applied to a model of the HSC network to construct strategies for controlling the differentiation of a stem cell population (Figure 2). D-SPIN also reveals how small molecules and small-molecule combinations can generate novel cell states and population structures (Figures 5-7), suggesting that D-SPIN could be used to study a broader set of interventions in both stem cells and immune cell populations. In both cases, a major goal is to control the distribution of cell states in a cell population, such as generating a specific set of progeny from stem cells or modulating the state of the immune system in cancer or autoimmunity. D-SPIN can uncover non-intuitive connections between different cellular processes, revealing how drugs might recruit a set of gene programs to generate a particular cell state. While this principle has been studied on a small scale for a small number of drugs [98], our results suggest that such principles of super-position might hold at the transcriptome scale.

In our analysis of drug responses, we demonstrate that drug combinations can act through coherent and incoherent superpositions to generate novel cell states. In both cases, drugs act in combination to recruit a combination of gene-expression programs within a single-cell type (Dasatinib and Halcinonide) or across a cell population (Motolimod and Tacrolimus). Superposition refers to the ability of drug combinations to function additively, resulting in the union of the gene-expression programs, each activated by the individual drugs. D-SPIN enables us to dissect these combinatorial mechanisms and interpret them through the inferred D-SPIN regulatory network and its interaction with drug response vectors. The results provide a conceptual framework for interpreting and predicting the effect of drug combinations. Although we have shown that additivity alone is insufficient to accurately predict combinatorial response due to the presence of non-additive interactions (Figure 7J), it could still serve as a guide to narrow down potential drug-combination targets for therapeutic objectives. Additivity could arise from the modularity of gene-regulatory circuits, such that different pathways impact gene expression levels independently. Further work is needed to reveal the specific conditions where additivity holds or breaks down [98].

The program-level or module-level description has been widely used in literature to provide interpretable insights into cellular regulation, but mainly from the view of the differential expression of single modules [43, 102, 103]. D-SPIN extends the program-level description by inferring interactions between programs. By representing the cell as a set of interacting programs, D-SPIN allows us to gain insight into the information flow between core cellular processes like transcription, translation, and protein degradation in response to single-gene knockdowns (Figure 4), as well as how therapeutic interventions combine additively to recruit modules of immune function (monocyte and M2 macrophage programs) to create novel cell states (Figure 7). How program-level regulation is implemented through biomolecular interactions is a future topic of research, which D-SPIN can facilitate as we show that D-SPIN can identify regulators of inhibition between mutually repressing modules by incorporating perturbations (Figure 2I).

The predictive power of D-SPIN and the connection between inferred interactions and biochemical interactions within the cell could be further enhanced by integrating data from other single-cell measurement modalities like ATAC-seq, known regulatory interactions, as well as predicted biochemical interactions. For example, single-cell ATAC-seq measures chromatin accessibility along the genome in single cells, providing information about the accessibility of transcription factor (TF) binding sites. Single-cell ATAC-seq data and transcription factor binding predictions can be combined with transcriptional profiling to identify TF-target interactions such as in methods like SCENIC+ and CellOracle [104, 105]. CellOracle in particular predicts the response of transcription factor regulatory networks to TF perturbations by integrating TF motif binding predictions with ATAC-seq data and linear regression. As D-SPIN is a statistical model, measured or predicted TF-target interactions can be included as a prior distribution during network inference, enhancing the model accuracy while still remaining flexible for biological contexts not dominantly controlled by TF-target interactions, such as signal transduction in T-cells. Further, techniques from deep neural networks and statistical learning have been used to learn representations and predict the result of perturbation experiments,

and future versions of D-SPIN could apply neural networks to construct energy functions that include higher-order interactions between more than two components and more realistic concentration-dependent regulatory functions [106, 107, 108, 109]. Incorporating more comprehensive terms into the energy function in Eq. 1 would enhance model fitting and predictive power, while the trade-off with model interpretability would be an important topic of further research.

D-SPIN's connection with spin network models provides insights into the fundamental nature of cellular regulation. The spin network model is an equilibrium model used to study physical systems at or near thermal equilibrium. Theoretically, D-SPIN depicts the cell population as a collection of points residing in an energy landscape that can be tilted by the perturbation vector to shift the distribution of cell states in a cell population. Energy-landscape models represent a highly simplified class of dynamical systems, as their behavior can be captured within a single energy (potential) function. Equilibrium spin network models have been used to study a much broader range of systems that are far from thermal equilibrium, including neural networks and bird flocks [36, 37, 31]. However, it remains unclear why equilibrium models can yield such significant predictive power for strongly non-equilibrium systems. The ability of equilibrium models to produce low-error reconstructions of cell population gene-expression states suggests that cells, in certain situations, may be effectively modeled as equilibrium systems driven through various configurations by a biasing drive. Such models have been demonstrated for cell-fate regulation and might represent a simplifying principle [10]. The ability to model a cell as an equilibrium system driven through different states presents a powerful simplification, as also explored in other chemical systems, offering a potential path toward more global theories of gene regulation [110, 111].

### Limitations of the Study

The D-SPIN framework has important current limitations that represent targets for extension in future work, although these possibilities fall beyond the scope of the present study. First, for simplicity and interpretability, D-SPIN only considers pairwise interactions between genes or gene programs in the network. These interactions correspond to the second-order terms in the energy function. For enhanced accuracy and predictive capabilities, the inclusion of higher-order multi-body interactions is a viable future modification. Second, D-SPIN is an equilibrium model and does not account for the dynamics of the system. Spin network models have natural extensions that incorporate dynamics, and studies of spin glass dynamics have been important in condensed matter physics [112, 113, 114]. Incorporating temporal dynamics would also allow the inclusion of directed edges, as directed network models in general cannot define consistent stationary distributions when the network contains feedback loops [115, 59, 116]. Third, D-SPIN assumes that the interactions within the core regulatory network $J$ are not altered by the perturbations, which is a reasonable approximation in perturbation response scenarios such as gene knockdown or small molecule action. However, in scenarios such as cellular differentiation or disease progression, the regulatory network may undergo changes under different conditions due to shifts in epigenetic regulation. Epigenetic reorganization can be included in future versions of D-SPIN by allowing interactions encoded in $J$ to also be condition-dependent.

## Author Contributions

Conceptualization, J.J., S.C., D.A.S., Z.J.G., M.T.; Methodology, J.J., D.A.S., M.T.; Data Analysis, J.J., S.C., C.S.M., J.H., M.T.; Experimental Investigation, S.C., T.T., C.S.M., T.K., J.H.P., J.V., E.D.C.; Data Curation, C.S.M., T.K., Q.Z., J.H., E.D.C.; Resources, S.C., T.T., C.S.M., T.K., Q.Z., J.V., J.H., E.D.C., Z.J.G., M.T.; Writing, J.J., S.C., D.A.S., M.T.; Visualization, J.J., S.C., I.S., M.T.; Software, J.J., Y.G.; Supervision, J.J., S.C., Z.J.G., M.T.; Funding Acquisition, Z.J.G., M.T.

## Supplementary Tables

Supplementary tables are available at Caltech Research Data Repository
`https://doi.org/10.22002/4b0jt-97c92`

## Data and code availability

Gene counts and metadata of drug profiling experiments are available at Caltech Research Data Repository `https://doi.org/10.22002/2cjss-wgh69`

Supplementary data of the analyses are available at Caltech Research Data Repository `https://doi.org/10.22002/bhvqz-qbs96`

MATLAB, Python implementations and Jupyter Notebook demonstrations of D-SPIN are available on GitHub `https://github.com/JialongJiang/DSPIN`

## Acknowledgements

We would like to acknowledge the NIH (TR01 GM150125, R01HD100039), the Heritage Medical Research Institute, Charles Trimble, the Shurl and Kay Curci Foundation, the Merkin Institute for Translational Research, 10x Genomics and Amgen. The single-cell profiling experiments were performed at the Beckman Institute Single-cell Profiling and Engineering Center (SPEC). We acknowledge the Caltech Protein Expression Center at the Beckman Institute of the California Institute of Technology for the use of the liquid handling robot. Sequencing was performed at the UCSF CAT, supported by UCSF PBBR, RRP IMIA, and NIH 1S10OD028511-01 grants. We acknowledge Dr. Guy Riddihough and Dr. Ariane Helou for editorial assistance with the manuscript. DAS acknowledges the support of an NSERC Discovery Grant and a Tier-II Canada Research Chair. We acknowledge Rob Phillips and Venkat Chandrasekaran for insightful scientific discussions. JH acknowledges the support of NIH 5R01GM135337 grants.

## Declaration of Interests

The work was supported in part by funds from a Caltech Amgen Research Collaboration Award and reagent gifts from 10x Genomics. MWT received funding from Adaptive Biotechnologies for unrelated work. MWT is a member of the advisory board of Cell Systems. MWT is a co-founder of CognitiveAI and YurtsAI. ZJG is a co-founder of Scribe Biosciences. The Regents of the University of California with ZJG and CSM as inventors have filed patent applications related to MULTI-seq.

## References

[1] Tabula Sapiens Consortium*, Robert C Jones, Jim Karkanias, Mark A Krasnow, Angela Oliveira Pisco, Stephen R Quake, Julia Salzman, Nir Yosef, Bryan Bulthaup, Phillip Brown, et al. The tabula sapiens: A multiple-organ, single-cell transcriptomic atlas of humans. *Science*, 376(6594):eabl4896, 2022.

[2] Eric H Davidson. *The regulatory genome: gene regulatory networks in development and evolution.* Elsevier, 2010.

[3] Dennis Bray. Protein molecules as computational elements in living cells. *Nature*, 376(6538):307–312, 1995.

[4] Aviv Regev and Ehud Shapiro. The $\pi$-calculus as an abstraction for biomolecular systems. In *Modelling in Molecular Biology*, pages 219–266. Springer, 2004.

[5] Aviv Regev and Ehud Shapiro. Cellular abstractions: Cells as computation. *Nature*, 419(6905):343–343, 2002.

[6] Eric H Davidson. Emerging properties of animal gene regulatory networks. *Nature*, 468(7326):911–920, 2010.

[7] Eric H Davidson and Douglas H Erwin. Gene regulatory networks and the evolution of animal body plans. *Science*, 311(5762):796–800, 2006.

[8] Lacramioara Bintu, Nicolas E Buchler, Hernan G Garcia, Ulrich Gerland, Terence Hwa, Jané Kondev, and Rob Phillips. Transcriptional regulation by the numbers: models. *Current opinion in genetics & development*, 15(2):116–124, 2005.

[9] David A Sivak and Matt Thomson. Environmental statistics and optimal regulation. *PLoS computational biology*, 10(9):e1003826, 2014.

[10] Cameron Sokolik, Yanxia Liu, David Bauer, Jade McPherson, Michael Broeker, Graham Heimberg, Lei S Qi, David A Sivak, and Matt Thomson. Transcription factor competition allows embryonic stem cells to distinguish authentic signals from noise. *Cell systems*, 1(2):117–129, 2015.

[11] Hao Yuan Kueh, Ameya Champhekar, Stephen L Nutt, Michael B Elowitz, and Ellen V Rothenberg. Positive feedback between pu. 1 and the cell cycle controls myeloid differentiation. *Science*, 341(6146):670–673, 2013.

[12] Nir Yosef, Alex K Shalek, Jellert T Gaublomme, Hulin Jin, Youjin Lee, Amit Awasthi, Chuan Wu, Katarzyna Karwacz, Sheng Xiao, Marsela Jorgolli, et al. Dynamic regulatory network controlling th17 cell differentiation. *Nature*, 496(7446):461–468, 2013.

[13] Jacques Monod. The growth of bacterial cultures. *Annual review of microbiology*, 3(1):371–394, 1949.

[14] Arthur B Pardee, François Jacob, and Jacques Monod. The genetic control and cytoplasmic expression of "inducibility" in the synthesis of $\beta$-galactosidase by e. coli. *Journal of Molecular Biology*, 1(2):165–178, 1959.

[15] 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, 2002.

[16] Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

[17] Günter P Wagner, Mihaela Pavlicev, and James M Cheverud. The road to modularity. *Nature Reviews Genetics*, 8(12):921–931, 2007.

[18] Tong Ihn Lee, Nicola J Rinaldi, François Robert, Duncan T Odom, Ziv Bar-Joseph, Georg K Gerber, Nancy M Hannett, Christopher T Harbison, Craig M Thompson, Itamar Simon, et al. Transcriptional regulatory networks in saccharomyces cerevisiae. *science*, 298(5594):799–804, 2002.

[19] Denes Hnisz, Brian J Abraham, Tong Ihn Lee, Ashley Lau, Violaine Saint-André, Alla A Sigova, Heather A Hoke, and Richard A Young. Super-enhancers in the control of cell identity and disease. *Cell*, 155(4):934–947, 2013.

[20] William T Ireland, Suzannah M Beeler, Emanuel Flores-Bautista, Nicholas S McCarty, Tom Röschinger, Nathan M Belliveau, Michael J Sweredoski, Annie Moradian, Justin B Kinney, and Rob Phillips. Deciphering the regulatory genome of escherichia coli, one hundred promoters at a time. *Elife*, 9:e55308, 2020.

[21] Christopher T Harbison, D Benjamin Gordon, Tong Ihn Lee, Nicola J Rinaldi, Kenzie D Macisaac, Timothy W Danford, Nancy M Hannett, Jean-Bosco Tagne, David B Reynolds, Jane Yoo, et al. Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431(7004):99–104, 2004.

[22] Sebastian J Maerkl and Stephen R Quake. A systems approach to measuring the binding energy landscapes of transcription factors. *Science*, 315(5809):233–237, 2007.

[23] Zhanzhi Hu, Patrick J Killion, and Vishwanath R Iyer. Genetic reconstruction of a functional transcriptional regulatory network. *Nature genetics*, 39(5):683–687, 2007.

[24] Linda S Huang and Paul W Sternberg. Genetic dissection of developmental pathways. *Methods in cell biology*, 48:97–122, 1995.

[25] Edwin L Ferguson, Paul W Sternberg, and H Robert Horvitz. A genetic pathway for the specification of the vulval cell lineages of caenorhabditis elegans. *Nature*, 326(6110):259–267, 1987.

[26] Chiou-Hwa Yuh, Hamid Bolouri, and Eric H Davidson. Genomic cis-regulatory logic: experimental and computational analysis of a sea urchin gene. *Science*, 279(5358):1896–1902, 1998.

[27] Daniel Schraivogel, Andreas R Gschwind, Jennifer H Milbank, Daniel R Leonce, Petra Jakob, Lukas Mathur, Jan O Korbel, Christoph A Merten, Lars Velten, and Lars M Steinmetz. Targeted perturb-seq enables genome-scale genetic screens in single cells. *Nature methods*, 17(6):629–635, 2020.

[28] 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, et al. Mapping information-rich genotype-phenotype landscapes with genome-scale perturb-seq. *Cell*, 2022.

[29] Christopher S McGinnis, David M Patterson, Juliane Winkler, Daniel N Conrad, Marco Y Hein, Vasudha Srivastava, Jennifer L Hu, Lyndsay M Murrow, Jonathan S Weissman, Zena Werb, et al. Multi-seq: sample multiplexing for single-cell rna sequencing using lipid-tagged indices. *Nature methods*, 16(7):619–626, 2019.

[30] Jase Gehring, Jong Hwee Park, Sisi Chen, Matthew Thomson, and Lior Pachter. Highly multiplexed single-cell rna-seq by dna oligonucleotide tagging of cellular proteins. *Nature Biotechnology*, 38(1):35–38, 2020.

[31] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[32] Edwin T Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.

[33] Alex H Lang, Hu Li, James J Collins, and Pankaj Mehta. Epigenetic landscapes explain partially reprogrammed cells and identify key reprogramming genes. *PLoS computational biology*, 10(8):e1003734, 2014.

[34] Andrew E Teschendorff and Andrew P Feinberg. Statistical mechanics meets single-cell biology. *Nature Reviews Genetics*, 22(7):459–476, 2021.

[35] Michele Castellana and William Bialek. Inverse spin glass and related maximum entropy problems. *Physical review letters*, 113(11):117204, 2014.

[36] Elad Schneidman, Michael J Berry, Ronen Segev, and William Bialek. Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, 440(7087):1007–1012, 2006.

[37] William Bialek, Andrea Cavagna, Irene Giardina, Thierry Mora, Edmondo Silvestri, Massimiliano Viale, and Aleksandra M Walczak. Statistical mechanics for natural flocks of birds. *Proceedings of the National Academy of Sciences*, 109(13):4786–4791, 2012.

[38] Narayana P Santhanam and Martin J Wainwright. Information-theoretic limits of selecting binary graphical models in high dimensions. *IEEE Transactions on Information Theory*, 58(7):4117–4134, 2012.

[39] 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*, pages 127–135, 2000.

[40] Dana Pe er, Aviv Regev, Gal Elidan, and Nir Friedman. Inferring subnetworks from perturbed expression profiles. *BIOINFORMATICS-OXFORD-*, 17:S215–S224, 2001.

[41] Wen Zhou, Mary A Yui, Brian A Williams, Jina Yun, Barbara J Wold, Long Cai, and Ellen V Rothenberg. Single-cell analysis reveals regulatory gene expression dynamics leading to lineage commitment in early t cell development. *Cell systems*, 9(4):321–337, 2019.

[42] Assen Roguev, Sourav Bandyopadhyay, Martin Zofall, Ke Zhang, Tamas Fischer, Sean R Collins, Hongjing Qu, Michael Shales, Han-Oh Park, Jacqueline Hayles, et al. Conservation and rewiring of functional modules revealed by an epistasis map in fission yeast. *science*, 322(5900):405–410, 2008.

[43] Eran Segal, Michael Shapira, Aviv Regev, Dana Pe'er, David Botstein, Daphne Koller, and Nir Friedman. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature genetics*, 34(2):166–176, 2003.

[44] Xiaoqiao Chen, Sisi Chen, and Matt Thomson. Minimal gene set discovery in single-cell mrna-seq datasets with activesvm. *Nature Computational Science*, 2(6):387–398, 2022.

[45] Seungjin Choi. Algorithms for orthogonal nonnegative matrix factorization. In *2008 ieee international joint conference on neural networks (ieee world congress on computational intelligence)*, pages 1828–1832. IEEE, 2008.

[46] Stephen A Vavasis. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, 20(3):1364–1377, 2010.

[47] Graham Heimberg, Rajat Bhatnagar, Hana El-Samad, and Matt Thomson. Low dimensionality in gene expression data enables the accurate extraction of transcriptional programs from shallow sequencing. *Cell systems*, 2(4):239–250, 2016.

[48] Sisi Chen, Paul Rivaud, Jong H Park, Tiffany Tsou, Emeric Charles, John R Haliburton, Flavia Pichiorri, and Matt Thomson. Dissecting heterogeneous cell populations across drug and disease conditions with popalign. *Proceedings of the National Academy of Sciences*, 117(46):28784–28794, 2020.

[49] Grace Hui Ting Yeo, Sachit D Saksena, and David K Gifford. Generative modeling of single-cell time series with prescient enables prediction of cell trajectories with interventions. *Nature communications*, 12(1):1–12, 2021.

[50] Caleb Weinreb, Samuel Wolock, Betsabeh K Tusi, Merav Socolovsky, and Allon M Klein. Fundamental limits on dynamic inference from single-cell snapshots. *Proceedings of the National Academy of Sciences*, 115(10):E2467–E2476, 2018.

[51] Jialong Jiang, David A Sivak, and Matt Thomson. Active learning of spin network models. *arXiv preprint arXiv:1903.10474*, 2019.

[52] Tara Chari and Lior Pachter. The specious art of single-cell genomics. *PLOS Computational Biology*, 19(8):e1011288, 2023.

[53] Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):192–225, 1974.

[54] Erik Aurell and Magnus Ekeberg. Inverse ising inference using all the data. *Physical review letters*, 108(9):090201, 2012.

[55] Pradeep Ravikumar, Martin J Wainwright, and John D Lafferty. High-dimensional ising model selection using $\ell_1$-regularized logistic regression. *The Annals of Statistics*, pages 1287–1319, 2010.

[56] H Chau Nguyen, Riccardo Zecchina, and Johannes Berg. Inverse statistical problems: from the inverse ising problem to data science. *Advances in Physics*, 66(3):197–261, 2017.

[57] Aditya Pratapa, Amogh P Jalihal, Jeffrey N Law, Aditya Bharadwaj, and TM Murali. Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nature methods*, 17(2):147–154, 2020.

[58] Jan Krumsiek, Carsten Marr, Timm Schroeder, and Fabian J Theis. Hierarchical differentiation of myeloid progenitors is encoded in the transcription factor network. *PloS one*, 6(8):e22649, 2011.

[59] Judea Pearl. Reverend bayes on inference engines: A distributed hierarchical approach. In *Probabilistic and Causal Inference: The Works of Judea Pearl*, pages 129–138. 2022.

[60] Leland H Hartwell, John J Hopfield, Stanislas Leibler, and Andrew W Murray. From molecular to modular cell biology. *Nature*, 402(Suppl 6761):C47–C52, 1999.

[61] Mark Newman. *Networks*. Oxford university press, 2018.

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

[63] Ryan N Gutenkunst, Joshua J Waterfall, Fergal P Casey, Kevin S Brown, Christopher R Myers, and James P Sethna. Universally sloppy parameter sensitivities in systems biology models. *PLoS computational biology*, 3(10):e189, 2007.

[64] Benjamin B Machta, Ricky Chachra, Mark K Transtrum, and James P Sethna. Parameter space compression underlies emergent theories and predictive models. *Science*, 342(6158):604–607, 2013.

[65] Brad T Sherman, Ming Hao, Ju Qiu, Xiaoli Jiao, Michael W Baseler, H Clifford Lane, Tomozumi Imamichi, and Weizhong Chang. David: a web server for functional enrichment analysis and functional annotation of gene lists (2021 update). *Nucleic Acids Res*, 10, 2022.

[66] Maxim V Kuleshov, Matthew R Jones, Andrew D Rouillard, Nicolas F Fernandez, Qiaonan Duan, Zichen Wang, Simon Koplev, Sherry L Jenkins, Kathleen M Jagodnik, Alexander Lachmann, et al. Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. *Nucleic acids research*, 44(W1):W90–W97, 2016.

[67] Damian Szklarczyk, Annika L Gable, Katerina C Nastou, David Lyon, Rebecca Kirsch, Sampo Pyysalo, Nadezhda T Doncheva, Marc Legeay, Tao Fang, Peer Bork, et al. The string database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic acids research*, 49(D1):D605–D612, 2021.

[68] Melanie de Almeida, Matthias Hinterndorfer, Hanna Brunner, Irina Grishkovskaya, Kashish Singh, Alexander Schleiffer, Julian Jude, Sumit Deswal, Robert Kalis, Milica Vunjak, et al. Akirin2 controls the nuclear import of proteasomes in vertebrates. *Nature*, 599(7885):491–496, 2021.

[69] Nora Heisterkamp, John R Stephenson, John Groffen, Pamela F Hansen, Annelies de Klein, Claus R Bartram, and Gerard Grosveld. Localization of the c-abl oncogene adjacent to a translocation break point in chronic myelocytic leukaemia. *Nature*, 306(5940):239–242, 1983.

[70] CB LOZZIO. Human chronic myelogenous leukemia cell-line with positive philadelphia chromosome. *Blood*, 45:321–334, 1975.

[71] Luz Garcia-Alonso, Christian H Holland, Mahmoud M Ibrahim, Denes Turei, and Julio Saez-Rodriguez. Benchmark and integration of resources for the estimation of human transcription factor activities. *Genome research*, 29(8):1363–1375, 2019.

[72] Zhi-Ping Liu, Canglin Wu, Hongyu Miao, and Hulin Wu. Regnetwork: an integrated database of transcriptional and post-transcriptional regulatory networks in human and mouse. *Database*, 2015:bav095, 2015.

[73] Heonjong Han, Jae-Won Cho, Sangyoung Lee, Ayoung Yun, Hyojin Kim, Dasom Bae, Sunmo Yang, Chan Yeong Kim, Muyoung Lee, Eunbeen Kim, et al. Trrust v2: an expanded reference database of human and mouse transcriptional regulatory interactions. *Nucleic acids research*, 46(D1):D380–D386, 2018.

[74] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[75] David Van Dijk, Roshan Sharma, Juozas Nainys, Kristina Yim, Pooja Kathail, Ambrose J Carr, Cassandra Burdziak, Kevin R Moon, Christine L Chaffer, Diwakar Pattabiraman, et al. Recovering gene interactions from single-cell data using data diffusion. *Cell*, 174(3):716–729, 2018.

[76] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):5233, 2019.

[77] Hyo-Ji Lee, Yunseo Woo, Tae-Wook Hahn, Young Mee Jung, and Yu-Jin Jung. Formation and maturation of the phagosome: a key mechanism in innate immunity against intracellular bacterial infection. *Microorganisms*, 8(9):1298, 2020.

[78] Pedro Beltrao, Gerard Cagney, and Nevan J Krogan. Quantitative genetic interactions reveal biological modularity. *Cell*, 141(5):739–745, 2010.

[79] Stefanie Jaeger, Peter Cimermancic, Natali Gulbahce, Jeffrey R Johnson, Kathryn E McGovern, Starlynn C Clarke, Michael Shales, Gaelle Mercenne, Lars Pache, Kathy Li, et al. Global landscape of hiv–human protein complexes. *Nature*, 481(7381):365–370, 2012.

[80] David E Gordon, Gwendolyn M Jang, Mehdi Bouhaddou, Jiewei Xu, Kirsten Obernier, Kris M White, Matthew J O'Meara, Veronica V Rezelj, Jeffrey Z Guo, Danielle L Swaney, et al. A sars-cov-2 protein interaction map reveals targets for drug repurposing. *Nature*, 583(7816):459–468, 2020.

[81] Minkyu Kim, Jisoo Park, Mehdi Bouhaddou, Kyumin Kim, Ajda Rojc, Maya Modak, Margaret Soucheray, Michael J McGregor, Patrick O'Leary, Denise Wolf, et al. A protein interaction landscape of breast cancer. *Science*, 374(6563):eabf3066, 2021.

[82] Belinda Wang, Rasika Vartak, Yefim Zaltsman, Zun Zar Chi Naing, Kelsey M. Hennick, Benjamin J. Polacco, Ali Bashir, Manon Eckhardt, Mehdi Bouhaddou, Jiewei Xu, Nawei Sun, Micaela C. Lasser, Yuan Zhou, Justin McKetney, Keelan Z. Guiley, Una Chan, Julia A. Kaye, Nishant Chadha, Merve Cakir, Martin Gordon, Prachi Khare, Sam Drake, Vanessa Drury, David F. Burke, Silvano Gonzalez, Sahar Alkhairy, Reuben Thomas, Stephanie Lam, Montana Morris, Ethel Bader, Meghan Seyler, Tierney Baum, Rebecca Krasnoff, Sheng Wang, Presley Pham, Juan Arbalaez, Dexter Pratt, Shivali Chag, Nadir Mahmood, Thomas Rolland, Thomas Bourgeron, Steven Finkbeiner, Danielle L. Swaney, Sourav Bandyopadhay, Trey Ideker, Pedro Beltrao, Helen Rankin Willsey, Kirsten Obernier, Tomasz J. Nowakowski, Ruth Hüttenhain, Matthew W. State, A. Jeremy Willsey, and Nevan J. Krogan. A foundational atlas of autism protein interactions reveals molecular convergence. *bioRxiv*, 2024.

[83] Joseph Hiatt, Judd F Hultquist, Michael J McGregor, Mehdi Bouhaddou, Ryan T Leenay, Lacy M Simons, Janet M Young, Paige Haas, Theodore L Roth, Victoria Tobin, et al. A functional map of hiv-host interactions in primary human t cells. *Nature communications*, 13(1):1752, 2022.

[84] Madalina Giurgiu, Julian Reinhard, Barbara Brauner, Irmtraud Dunger-Kaltenbach, Gisela Fobo, Goar Frishman, Corinna Montrone, and Andreas Ruepp. Corum: the comprehensive resource of mammalian protein complexes—2019. *Nucleic acids research*, 47(D1):D559–D563, 2019.

[85] Maria Rosaria Amoroso, Danilo Swann Matassa, Gabriella Laudiero, Anastasia V Egorova, Roman S Polishchuk, Francesca Maddalena, Annamaria Piscazzi, Simona Paladino, Daniela Sarnataro, Corrado Garbi, et al. Trap1 and the proteasome regulatory particle tbp7/rpt3 interact in the endoplasmic reticulum and control cellular ubiquitination of specific mitochondrial proteins. *Cell Death & Differentiation*, 19(4):592–604, 2012.

[86] Chiara Milanese, Cíntia R Bombardieri, Sara Sepe, Sander Barnhoorn, César Payán-Goméz, Donatella Caruso, Matteo Audano, Silvia Pedretti, Wilbert P Vermeij, Renata MC Brandt, et al. Dna damage and transcription stress cause atp-mediated redesign of metabolism and potentiation of anti-oxidant buffering. *Nature communications*, 10(1):4887, 2019.

[87] Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058, 2018.

[88] Yvan Jamilloux, Thomas El Jammal, Lucine Vuitton, Mathieu Gerfaud-Valentin, Sébastien Kerever, and Pascal Sève. Jak inhibitors for the treatment of autoimmune and inflammatory diseases. *Autoimmunity reviews*, 18(11):102390, 2019.

[89] David C Fajgenbaum and Carl H June. Cytokine storm. *New England Journal of Medicine*, 383(23):2255–2273, 2020.

[90] Lisa Holthaus, Daniel Lamp, Anita Gavrisan, Virag Sharma, Anette-Gabriele Ziegler, Martin Jastroch, and Ezio Bonifacio. Cd4+ t cell activation, function, and metabolism are inhibited by low concentrations of dmso. *Journal of immunological methods*, 463:54–60, 2018.

[91] C Domínguez Conde, C Xu, LB Jarvis, DB Rainbow, SB Wells, T Gomes, SK Howlett, O Suchanek, K Polanski, HW King, et al. Cross-tissue immune cell analysis reveals tissue-specific features in humans. *Science*, 376(6594):eabl5197, 2022.

[92] Ctibor Skuta, Martin Popr, Tomas Muller, Jindrich Jindrich, Michal Kahle, David Sedlak, Daniel Svozil, and Petr Bartunek. Probes & drugs portal: an interactive, open data resource for chemical biology. *Nature methods*, 14(8):759–760, 2017.

[93] David S Wishart, Craig Knox, An Chi Guo, Savita Shrivastava, Murtaza Hassanali, Paul Stothard, Zhan Chang, and Jennifer Woolsey. Drugbank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic acids research*, 34(suppl_1):D668–D672, 2006.

[94] Daniella M Schwartz, Yuka Kanno, Alejandro Villarino, Michael Ward, Massimo Gadina, and John J O'Shea. Jak inhibition as a therapeutic strategy for immune and inflammatory diseases. *Nature reviews Drug discovery*, 16(12):843–862, 2017.

[95] Bram Van de Sande, Joon Sang Lee, Euphemia Mutasa-Gottgens, Bart Naughton, Wendi Bacon, Jonathan Manning, Yong Wang, Jack Pollard, Melissa Mendez, Jon Hill, et al. Applications of single-cell rna sequencing in drug discovery and development. *Nature Reviews Drug Discovery*, pages 1–25, 2023.

[96] Thibaut Desgeorges, Giorgio Caratti, Rémi Mounier, Jan Tuckermann, and Bénédicte Chazaud. Glucocorticoids shape macrophage phenotype for tissue repair. *Frontiers in immunology*, 10:1591, 2019.

[97] Yan-Cun Liu, Xian-Biao Zou, Yan-Fen Chai, and Yong-Ming Yao. Macrophage polarization in inflammatory diseases. *International journal of biological sciences*, 10(5):520, 2014.

[98] Naama Geva-Zatorsky, Erez Dekel, Ariel A Cohen, Tamar Danon, Lydia Cohen, and Uri Alon. Protein dynamics in drug combinations: a linear superposition of individual-drug responses. *Cell*, 140(5):643–651, 2010.

[99] JA Fishman. Infection in organ transplantation. *American Journal of Transplantation*, 17(4):856–879, 2017.

[100] Jay A Fishman and Robert H Rubin. Infection in organ-transplant recipients. *New England Journal of Medicine*, 338(24):1741–1751, 1998.

[101] Marcus R Pereira, Sumit Mohan, David J Cohen, Syed A Husain, Geoffrey K Dube, Lloyd E Ratner, Selim Arcasoy, Meghan M Aversa, Luke J Benvenuto, Darshana M Dadhania, et al. Covid-19 in solid organ transplant recipients: initial report from the us epicenter. *American journal of transplantation*, 20(7):1800–1808, 2020.

[102] Eran Segal, Nir Friedman, Daphne Koller, and Aviv Regev. A module map showing conditional activity of expression modules in cancer. *Nature genetics*, 36(10):1090–1098, 2004.

[103] Ido Amit, Manuel Garber, Nicolas Chevrier, Ana Paula Leite, Yoni Donner, Thomas Eisenhaure, Mitchell Guttman, Jennifer K Grenier, Weibo Li, Or Zuk, et al. Unbiased reconstruction of a mammalian transcriptional network mediating pathogen responses. *Science*, 326(5950):257–263, 2009.

[104] Carmen Bravo González-Blas, Seppe De Winter, Gert Hulselmans, Nikolai Hecker, Irina Matetovici, Valerie Christiaens, Suresh Poovathingal, Jasper Wouters, Sara Aibar, and Stein Aerts. Scenic+: single-cell multiomic inference of enhancers and gene regulatory networks. *Nature methods*, 20(9):1355–1367, 2023.

[105] Kenji Kamimoto, Blerta Stringa, Christy M Hoffmann, Kunal Jindal, Lilianna Solnica-Krezel, and Samantha A Morris. Dissecting cell identity via network inference and in silico gene perturbation. *Nature*, 614(7949):742–751, 2023.

[106] Mohammad Lotfollahi, F Alexander Wolf, and Fabian J Theis. scgen predicts single-cell perturbation responses. *Nature methods*, 16(8):715–721, 2019.

[107] Mohammad Lotfollahi, Anna Klimovskaia Susmelj, Carlo De Donno, Leon Hetzel, Yuge Ji, Ignacio L Ibarra, Sanjay R Srivatsan, Mohsen Naghipourfar, Riza M Daza, Beth Martin, et al. Predicting cellular responses to complex perturbations in high-throughput screens. *Molecular Systems Biology*, page e11517, 2023.

[108] Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with gears. *Nature Biotechnology*, pages 1–9, 2023.

[109] Mingze Dong, Bao Wang, Jessica Wei, Antonio H de O. Fonseca, Curtis J Perry, Alexander Frey, Feriel Ouerghi, Ellen F Foxman, Jeffrey J Ishizuka, Rahul M Dhodapkar, et al. Causal identification of single-cell experimental perturbation effects with cinema-ot. *Nature Methods*, 20(11):1769–1779, 2023.

[110] Lacramioara Bintu, Nicolas E Buchler, Hernan G Garcia, Ulrich Gerland, Terence Hwa, Jane Kondev, Thomas Kuhlman, and Rob Phillips. Transcriptional regulation by the numbers: applications. *Current opinion in genetics & development*, 15(2):125–135, 2005.

[111] Jeremy Gunawardena. A linear framework for time-scale separation in nonlinear biochemical systems. *PloS one*, 7(5):e36321, 2012.

[112] Haim Sompolinsky and Annette Zippelius. Relaxational dynamics of the edwards-anderson model and the mean-field theory of spin-glasses. *Physical Review B*, 25(11):6860, 1982.

[113] Daniel S Fisher and David A Huse. Nonequilibrium dynamics of spin glasses. *Physical Review B*, 38(1):373, 1988.

[114] Roy J Glauber. Time-dependent statistics of the ising model. *Journal of mathematical physics*, 4(2):294–307, 1963.

[115] Judea Pearl. Evidential reasoning using stochastic simulation of causal models. *Artificial intelligence*, 32(2):245–257, 1987.

[116] Barry C Arnold and S James Press. Compatible conditional distributions. *Journal of the American Statistical Association*, 84(405):152–156, 1989.

# STAR Methods

## Contents


**1 Method details** **14**
1.1 Overview of D-SPIN framework . . . 14
1.2 Formulation of D-SPIN framework . . . 15
1.3 Computational methods for inference in D-SPIN . . . 16
1.4 Regularization of the inference . . . 18
1.5 Proof of convexity of the inference problem . . . 18
1.6 Quantification of the inference uncertainty . . . 19
1.7 Correspondence between D-SPIN and biochemical reaction systems . . . 20
1.8 Gene program discovery by oNMF . . . 21
1.9 Evaluating gene program expression and discretization . . . 22
1.10 Selection of gene program number in oNMF . . . 22
1.11 High-throughput drug perturbation experiments on PBMCs . . . 23
1.12 High-throughput single-cell profiling using sample multiplexing techniques . . . 23
1.13 Time-course profiling of T-cell mediated immune activation . . . 24
1.14 Alignment and de-multiplexing . . . 24

**2 Quantification and statistical analysis** **24**
2.1 Filtering, normalization, and clustering of the Perturb-seq dataset . . . 24
2.2 Filtering, normalization, and clustering of the drug profiling dataset . . . 25
2.3 oNMF decomposition of single-cell dataset . . . 26
2.4 Inferring regulatory networks and perturbation responses with D-SPIN . . . 26
2.5 Parallelization of D-SPIN inference . . . 27
2.6 Simulations of the HSC network . . . 27
2.7 Clamping analysis of node marginal distribution and edge sensitivity . . . 28
2.8 Simulation of large synthetic networks . . . 28
2.9 Benchmarking time consumption with cell number . . . 29
2.10 Evaluating single-gene network reconstruction with ChIP-seq data . . . 29
2.11 Discovering regulatory networks modules and clustering perturbation responses . . . 30
2.12 Automated identification of regulatory strategies and network coarse-graining . . . 31
2.13 Embedding and comparison of D-SPIN and data distributions . . . 31
2.14 Coherence of molecular complexes in CORUM . . . 32
2.15 Identification of drug target and potency from database . . . 33
2.16 Computing strength of immune inhibitors . . . 33
2.17 Categories of drug interactions on gene program level . . . 33
2.18 Coherence between drug responses . . . 34
2.19 Phase diagrams of dosages of drug combination . . . 34

# Supplementary Figures

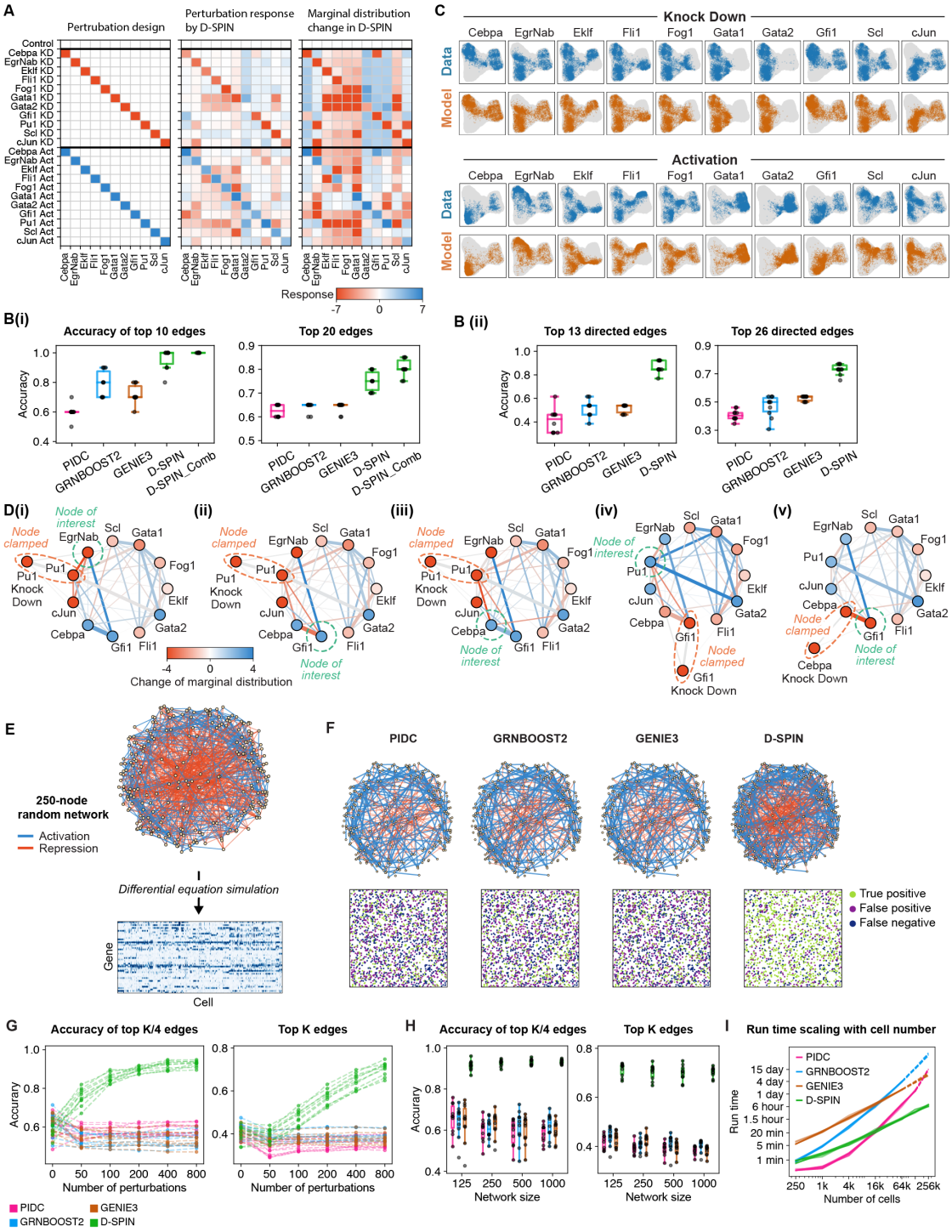


Figure S2: Caption on the following page

Figure S2 **Perturbation responses and generated cell state distributions by D-SPIN; D-SPIN scales to large networks with a thousand genes; D-SPIN has superior running-time-scaling with cell number; related to Figure 2** (A) Heatmaps indicating (left) applied perturbations to the network; (middle) perturbation response inferred by D-SPIN (right) marginal distribution changes of single genes as detailed in Sec. 2.7. The heatmaps demonstrate that the D-SPIN response vectors can identify the applied perturbations and quantify their impact on single genes. The only discrepancy of inferred perturbation occurs on Gata2 knockdown, potentially because Gata2 is a transient regulator in the network, having low expression in differentiated cell states. (B) (i) Bar plots quantify network inference accuracy of D-SPIN with combinatorial perturbations on top-predicted 10 and 20 edges. D-SPIN achieves further accuracy improvements with additional 96 double-gene perturbations, with exact top 10 predictions and 0.805 accuracy in the top 20 predictions. (ii) Bar plots quantify the directed network inference accuracy of D-SPIN on top-predicted 13 and 26 edges where the ground truth network has 26 directed edges. D-SPIN is capable of inferring a directed network, as detailed in Sec. 1.3, with superior accuracy (0.73 on top 26 edges) compared with existing methods (0.41, 0.47, 0.53 on top 26 edges). (C) UMAP embedding of cell state distributions of (blue) simulated data by BEELINE and (orange) D-SPIN model distributions across all single-gene knockdown/activation perturbations. The D-SPIN model accurately generates cell-state distributions with the regulatory network and each perturbation across all these conditions. (D) Network diagrams visualize edge sensitivity analysis in D-SPIN showing how each network edge positively (blue) or negatively (red) contributes to the marginal distribution change of the gene of interest under a clamped perturbation, as detailed in Sec. 2.7. Network nodes are colored by the upregulation or downregulation of their activities defined by marginal distributions, and edges are colored by their contribution to the activity of the node of interest. For example, panel (ii) shows under Pu1 knockdown, the upregulation of Gfi1 is most significantly positively contributed by the edge EgrNab-Gfi1. (E) (top) Network diagram of an example 250-node Erdős–Rényi (ER) network with equal probability of activating (blue) and inhibiting (red) edges. (bottom) Heatmap showing simulated single-cell data for evaluating the accuracy and scalability of D-SPIN as detailed in Sec. 2.8. (F) Diagrams of the inferred network of different methods as in (top) subnetwork of correct-inferred edges and (bottom) adjacency matrices with true positives, false positives, and false negatives. D-SPIN achieves superior accuracy compared to existing methods (G) The network inference accuracies are plotted with the number of perturbations in 10 randomly generated ER networks. The accuracy of D-SPIN continuously increases with perturbation number while other methods do not improve or slightly decrease. (H) Bar plots quantify the network inference accuracy of ER networks with different sizes of 125, 250, 500, and 1000 under 800 perturbations in 10 random repeats. D-SPIN achieves significantly increased accuracy compared with other methods across network sizes. (I) The running time of different network inference methods as a function of cell number in a 250-node ER network. D-SPIN only takes a few hours for 256,000 cells, while other methods require weeks. Dashed lines are interpolations, as detailed in Sec. 2.9.

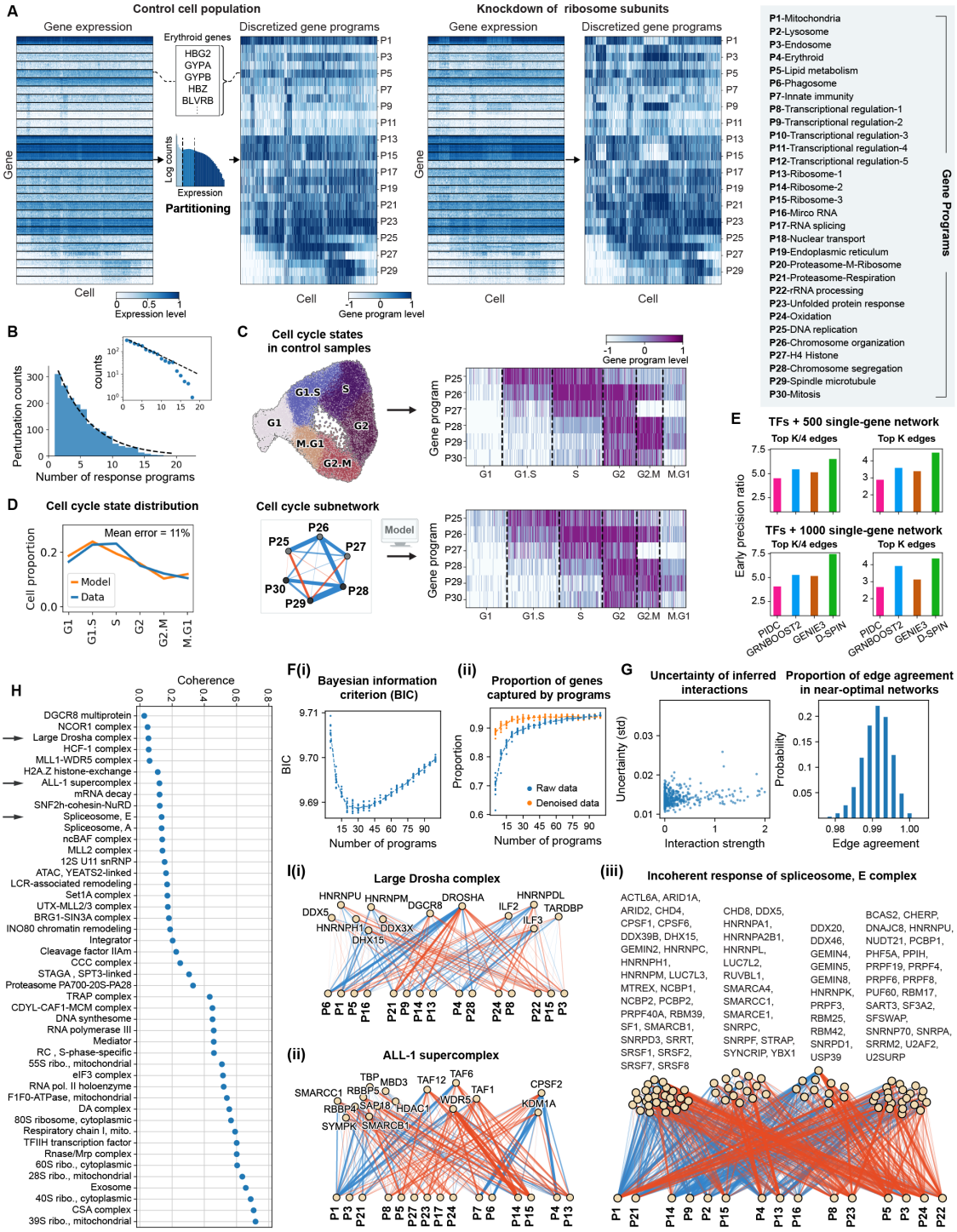


Figure S3: Caption on the following page

Figure S3 **Examples of gene program discretization; D-SPIN reconstructs cell cycle states; Evaluating single-gene network accuracy with ChIP-seq datasets; The choice of gene-program number; Uncertainty of the inferred network; Examples of incoherent macromolecular complexes; related to Figure 3** (A) Heatmaps of gene expression and discretized gene program level for (left) control samples of non-targeting guide RNAs and (right) knockdown of the large or small ribosome subunits. The expression of each gene is normalized by its maximum expression and capped at 0.8 for visualization. The heatmaps demonstrate that discretized gene programs characterize and denoise major expression patterns in the gene matrix. (right box) Gene programs are functionally annotated through gene ontology annotation tools [1, 2] and manual lookup. (B) Histogram showing the number of responding gene programs for each gene knockdown perturbation on a linear scale and (inset) log scale. The responding program number is exponentially distributed, suggesting that perturbations influencing a large number of perturbations are relatively rare. (C) (left-top) UMAP embedding of cell-cycle-related gene programs in control samples exhibits a circular structure. The clusters of cell cycle states are annotated and ordered based on the gene program expression and sequential positions on the UMAP. (left-bottom) Network diagram of the D-SPIN cell cycle network, which is taken as the subnetwork of 6 cell-cycle programs from the full 30-program network. (right-top) Heatmaps of cell-cycle programs for cell states from the Perturb-seq control samples and (right-bottom) the D-SPIN model under the control condition. The heatmaps demonstrate that D-SPIN reconstructs the distribution of cell states associated with cell cycle progression from the 6-node subnetwork. (D) The distribution of cell cycle states in generated data from the D-SPIN model and experimental Perturb-seq control samples. The distribution generated by D-SPIN quantitatively agrees with the data distribution with an 11% mean error. (E) Correspondence of D-SPIN and CHIP-seq data for single-gene network inference on two networks, TFs+500 and TFs+1000, derived from the Perturb-seq dataset. The correspondence is quantified early precision ratio, the fraction of top-inferred edges that can be associated with a known transcriptional factor (TF)-target binding data from ChIP-seq database compared with a random predictor, as detailed in Sec. 2.10. D-SPIN achieves better correspondence with ChIP-seq data on single-gene regulatory network inference compared with existing methods. (F)(i) Bayesian information criterion (BIC) of the gene program decomposition as a function of gene program number. 30 is the optimal program number that minimizes BIC when the effective sample size is 20k cells. (ii) The proportion of genes that have a high correlation with their corresponding gene programs as a function of gene program number. For raw data (without application of MAGIC), a high correlation is defined as exceeding 4 standard deviations of gene-gene correlation distribution. In the denoised data by MAGIC [3], a high correlation is defined as larger than 0.5. The plot shows that 30 programs is an elbow point of information gain, as detailed in Sec. 1.10. (G) (left) Uncertainty of inferred network interactions as a function of interaction strength quantified by the standard deviation among alternative network solutions, as detailed in Sec. 1.6. The uncertainty is generally below 0.02 and much lower compared with interaction strengths. (right) Histogram showing the proportion of edge agreement between alternative solutions of the D-SPIN network inference problem. Alternative solutions are obtained by sampling from the Bayesian posterior distribution given the experimental data, as detailed in Sec. 1.6. Alternative solutions of D-SPIN are highly consistent with the inferred network, with typically more than 98% of the edges being in the same category of activation, inhibition, and non-interacting. (H) Full plot of regulatory coherence of macromolecular complex subunits of the 46 filtered molecular complexes in the CORUM database [4]. In low-coherence complexes, knockdowns of different subunits have different interactions with the regulatory network, suggesting distinct functions for complex subunits. (I) Diagrams of response vectors for example incoherent molecular complexes that have signatures of sub-functionalization. The subunits of (i) large Drosha complex and (ii) ALL-1 supercomplex are partitioned into 3 groups. The subunits of (iii) spliceosome E complex are partitioned into 4 groups.

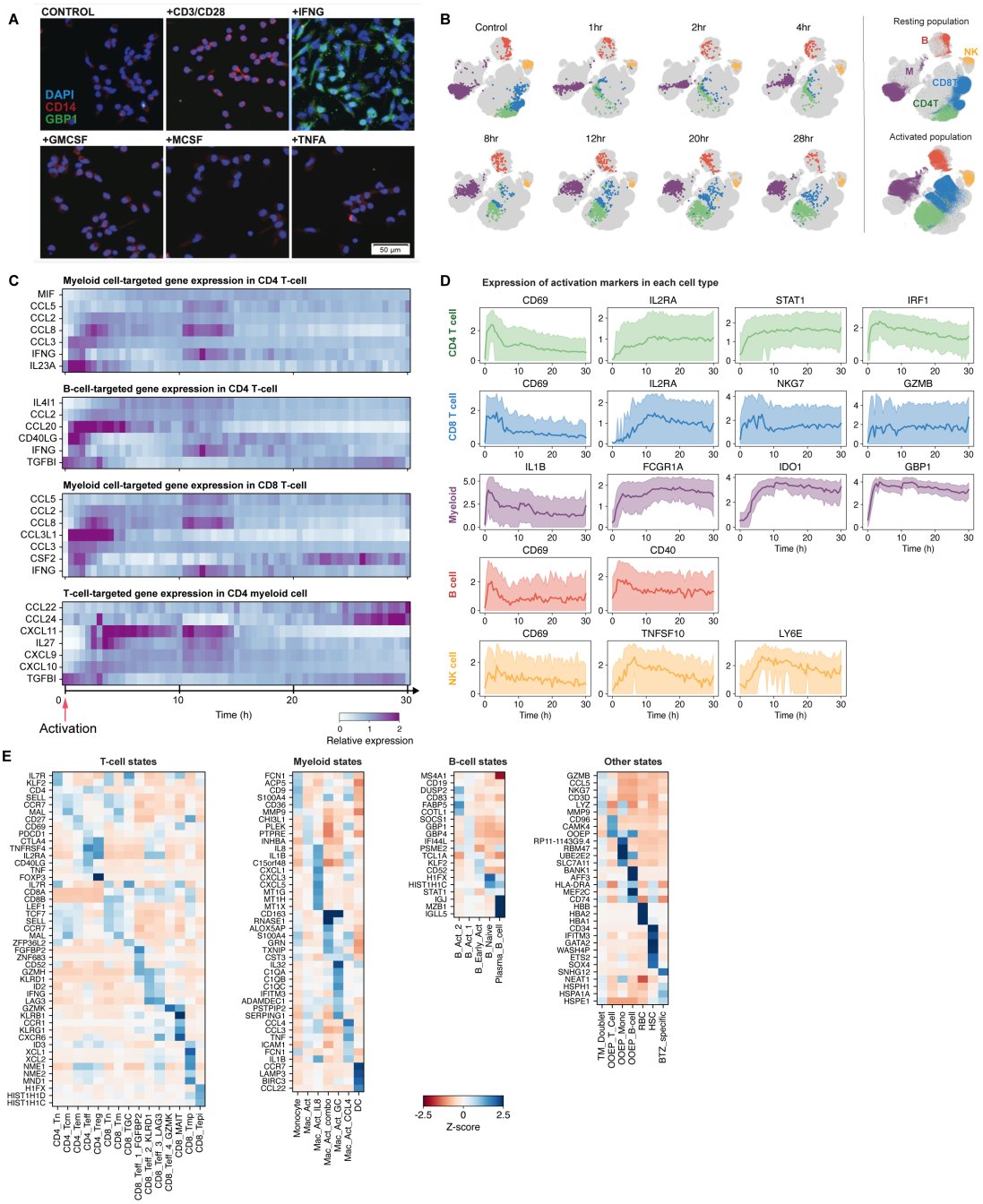


Figure S4: Caption on the following page

Figure S4 **Timecourse of T-cell mediated immune activation; cell typing of the drug profiling experiments; related to Figure 5** (A) Immunostaining of primary human monocytes cultured in CD3/CD28 antibody (25uL/mL cells), IFNG (5ng/mL), GMCSF (5ng/mL), MCSF (5ng/mL), or TNFA (5ng/mL). The monocyte activation marker GBP1 is only expressed under IFNG activation. The results show that anti-CD3/CD28 antibody does not activate monocytes alone, and the observed monocyte activation in the PBMC is due to signaling between cell types. (B) UMAP embeddings of example time points in the 30-hour time-course experiment of T-cell-mediated immune activation, where samples were taken every 30 minutes for single-cell mRNA-seq. The immune population gradually moves from the resting state to the activated state. (C) Signaling gene expression over time for select signaling genes in four cell types. Expression is normalized by the mean expression across the time course. Many signaling genes are immediately upregulated in T-cells after antibody activation, while in myeloid cells signaling genes such as IL27, and CXCL11 are upregulated after a few hours, showing rich dynamics of communications between different cell types. (D) The dynamics of gene markers for immune activation in each major cell type. The color-shaded range is the 10th and 90th percentile of gene expression. After T-cell activation by the anti-CD3/CD28 antibody, major cell types in PBMC are activated with different time dynamics. (E) The heatmap shows the Z-score of gene expression for cell type markers. Marker genes are identified by differential expression analysis of each Leiden cluster. The Z-score is computed from the mean marker gene expression in each Leiden cluster compared to the mean and standard deviation in all the cells of the same major cell type (T-cell, Myeloid, B-cell, and others).

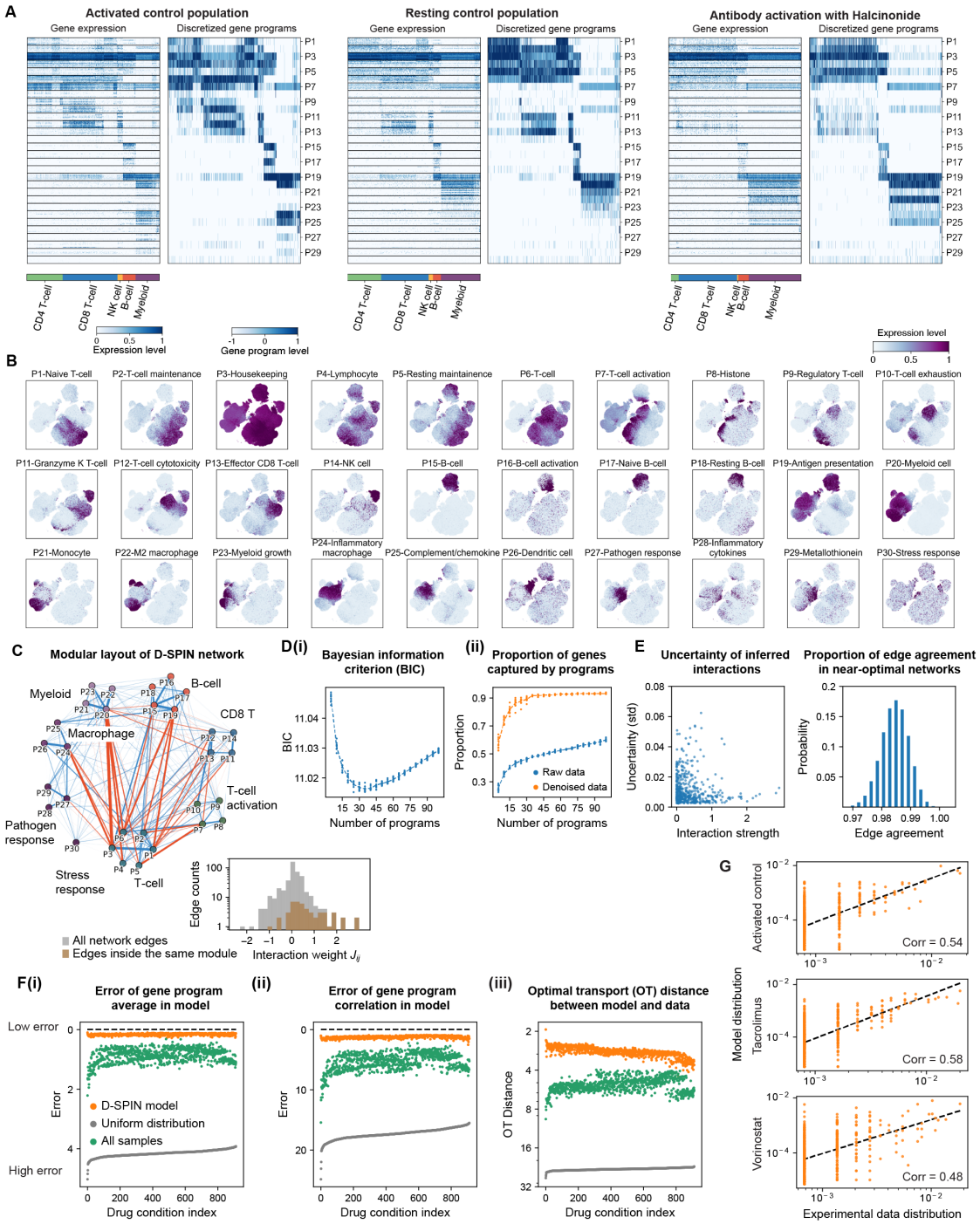


Figure S5: Caption on the following page

Figure S5 **Examples of gene program discretization and gene program visualization on immune cell population UMAP; The choice of gene-program number; Uncertainty of edges in the inferred network; Comparison of cell state distributions between D-SPIN model and experimental data; related to Figure 5** (A) Heatmaps of gene expression and discretized gene program level on (left) activated control population (middle) resting control population, and (right) T-cell receptor activation with Halcinonide treatment. The expression of each gene is normalized by its maximum expression and capped at 0.8 for visualization. The heatmaps demonstrate that discretized gene programs characterize and denoise major expression patterns in the gene matrix. (B) UMAP rendering of the expression level of each gene program. Each cell is colored by its expression level on the gene program. The expression level is normalized by the 95th percentile for visualization only. The plots demonstrate that gene programs are specifically expressed by certain cell states, which are localized in separate regions on the UMAP. (C) (top) Network diagram of the regulatory network model by D-SPIN exhibits modular structure with tightly connected subnetworks. The 8 network modules correspond to major cell types or cell type functions in the PBMC population, such as T-cells, B-cells, and myeloid cells. (bottom) The histogram quantifies the distribution of all network edges ($J$ matrix entries) and edges inside the same module. Edges inside modules are mostly positive interactions and contain the majority of strong positive interactions in the network. (D)(i) Bayesian information criterion (BIC) of the gene program decomposition as a function of gene program number. 30 is the optimal program number that minimizes BIC when the effective sample size is 15k cells. (ii) The proportion of genes that have a high correlation with their corresponding gene programs as a function of gene program number. In raw data, a high correlation is defined as exceeding 3 standard deviations of gene-gene correlation distribution. In the denoised data by MAGIC [3], a high correlation is defined as larger than 0.5. The plot shows that 30 programs is an elbow point of information gain, as detailed in Sec. 1.10. (E) (left) Uncertainty of inferred network interactions as a function of interaction strength quantified by the standard deviation among alternative network solutions, as detailed in Sec. 1.6. The uncertainty is generally below 0.06 and much lower compared with interaction strengths. (right) Histogram showing the proportion of edge agreement between alternative solutions of the D-SPIN network inference problem. Alternative solutions are obtained by sampling from the Bayesian posterior distribution given the experimental data, as detailed in Sec. 1.6. Alternative solutions of D-SPIN are highly consistent with the inferred network, with typically more than 97% of the edges being in the same category of activation, inhibition, and non-interacting. (F) Scatter plots of evaluating how the experimental data distribution is different from the D-SPIN model distribution and two reference null distributions by (i) Euclidean distance between the average of gene program expression, (ii) Frobenius norm of the difference between the cross-correlation of gene program expression, and (iii) Optimal transport distance of probability distributions. The two null distributions are (gray) uniform distribution over all states and (green) the distribution of pooling samples from all experimental conditions to reflect relative cell type abundance, as detailed in Sec. 2.13. Across all experimental conditions, the distributions defined by D-SPIN quantitatively agree better with the data distribution compared with the two reference distributions. (G) Scatter plot comparisons between state distributions of experimental data and the D-SPIN model. Even though the experimental observations are sparse samples of a few thousand cells in the total $3^{30} \approx 2 \times 10^{14}$ possible states, the probabilities of empirical distribution and D-SPIN model probabilities are moderately correlated with correlation 0.54, 0.58, and 0.48. The plot shows that high-probability states in the D-SPIN model are also high-probability states in experimental data.

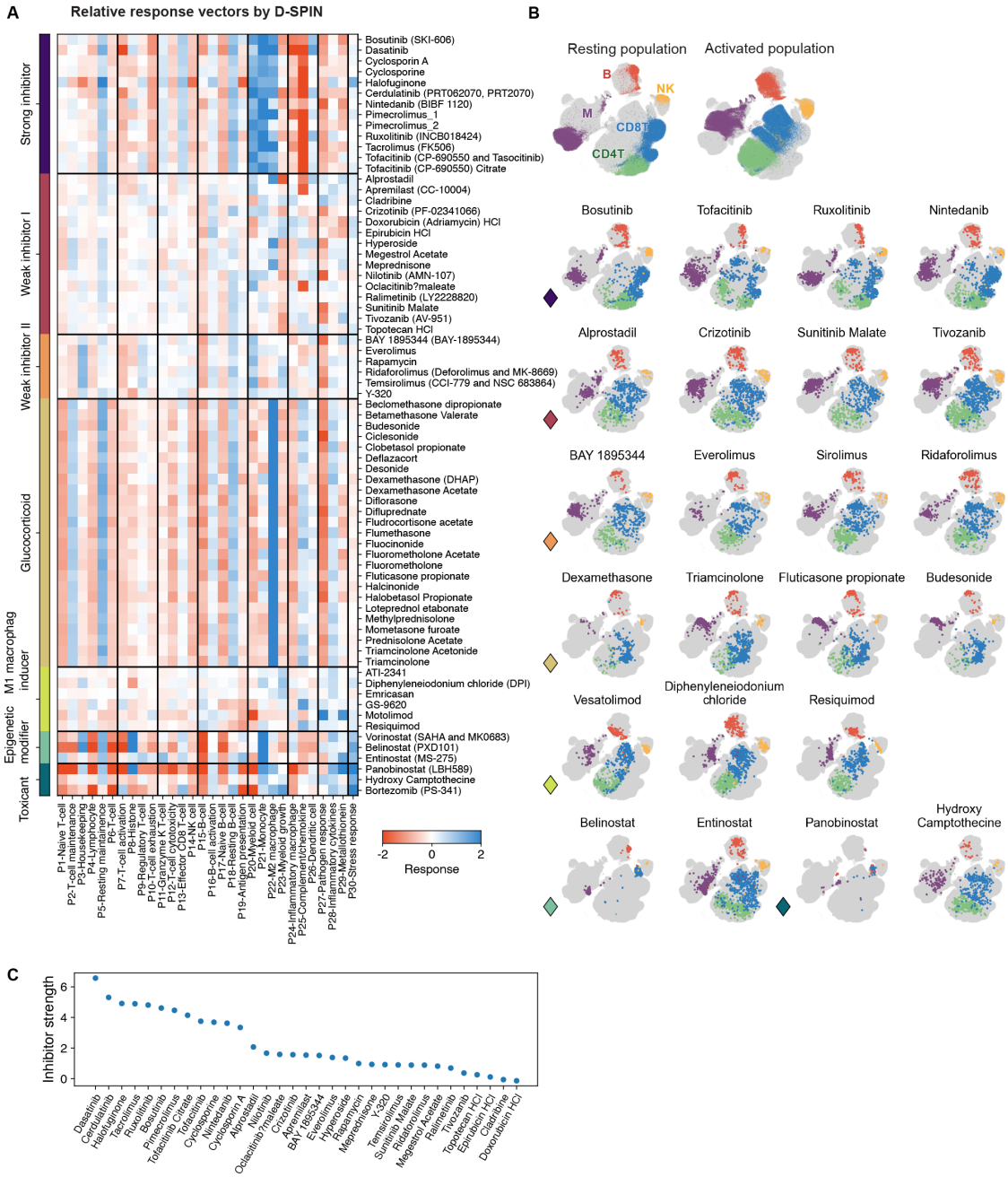


Figure S6: **D-SPIN identifies phenotypical classes of drug categories; related to Figure 6** (A) Heatmap of drug relative response vectors inferred by D-SPIN. The relative response vectors are defined as the change of drug response vector over control samples. D-SPIN identifies 70 effective drugs on the immune population that have statistically significantly different relative response vectors from control samples, as detailed in Sec. 2.11. We group these drugs into 7 phenotypical classes: strong inhibitor, weak inhibitor I, weak inhibitor II, glucocorticoid, M1 macrophage inducer, epigenetic modifier, and toxicant. (B) UMAP embeddings of example drugs from each class. Drugs in the same phenotypical class identified by D-SPIN exhibit similar cell population shifts on the UMAP embedding, while the cell population states vary greatly between different classes. (C) Strength of immune inhibition of all D-SPIN-identified immune inhibitors. Inhibitor strengths are computed by projecting the relative response vector towards the leading singular vector of all inhibitors, as detailed in Sec. 2.16.

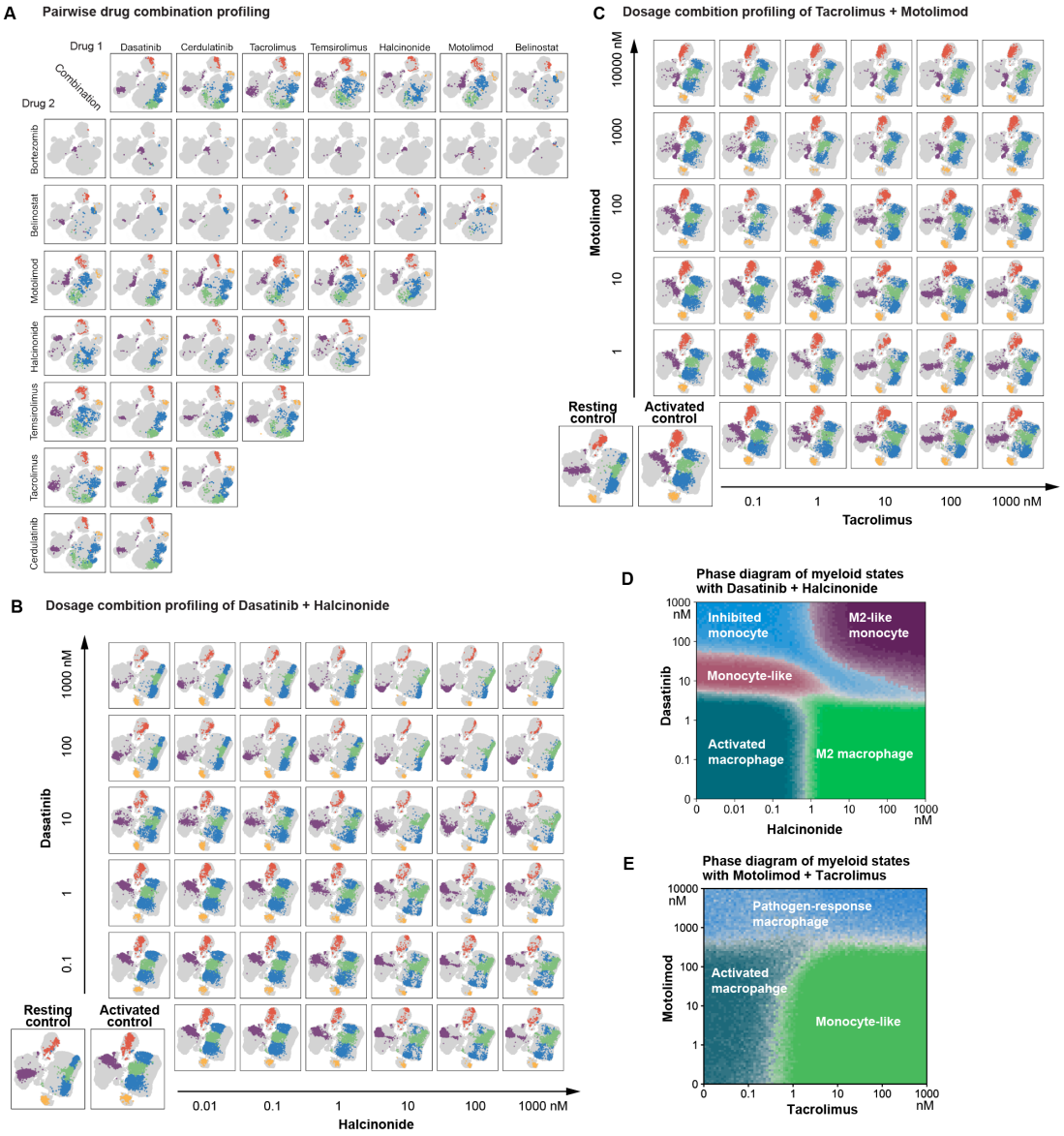


Figure S7: Caption on the following page

Figure S7 **Drug combination profiling discovers novel cell states; dosage combination reveals phase diagrams of drug combination effect; related to Figure 7** (A) UMAP embeddings of selected single drugs and pairwise drug combinations. We performed pair-wise drug combination profiling for 12 selected drugs (8 shown here) from different categories. (a). Strong inhibitor: broad-spectrum kinase inhibitor Dasatinib, BCR-ABL, and src tyrosine kinase inhibitor Bosutinib, SYK/JAK kinase inhibitor Cerdulatinib, calcineurin phosphatase inhibitor Tacrolimus; (b). Weak inhibitor: mTOR kinase inhibitor Temsirolimus; (c). Glucocorticoid: Halcinonide; (d). M1-macrophage inducer: Motolimod; (e). Epigenetic modifier: Vorinostat, Belinostat; (f). Toxicant: Bortezomib; (g). Others: Forsythin, Naproxen sodium. Forsythin and Naproxen cause the loss of macrophage population in single-drug profiling, which is not observed in repeats in drug combination profiling. (B) UMAP embeddings of dosage combination profiling of Dasatinib plus Halcinonide. The two drugs create a Monocyte-like resting myeloid population at a medium level of combination (10nM Dasatinib, 1nM Halcinonide) and a novel combinatorial myeloid state expressing both Monocyte and M2 macrophage gene programs. On T-cells, the drug combination also produces inhibited T-cell states. (C) UMAP embeddings of dosage combination profiling of Tacrolimus plus Motolimod. The drug combination produces myeloid states that are similar to Motolimod-induced pathogen-responding myeloid states, and intermediate T-cell states that have gene program expression signatures of both drugs. (D) The phase diagram of myeloid states under different drug dosage combinations of Dasatinib and Halcinonide depicted by D-SPIN. The phase diagram is computed by evaluating the likelihood of each reference cell state (activated macrophage, M2 macrophage, etc.) at different dosage combinations using spline interpolation of response vectors of profiled dosages as detailed in Sec. 2.19. At a low dosage of Dasatinib, the myeloid state transitions from the activated macrophage state to the M2 macrophage state as the Halcinonide dosage increases. At high dosages of Dasatinib, increasing Halcinonide dosage produces Monocyte-like, inhibited monocyte, and M2-like monocyte states. (E) The phase diagram of myeloid states under different drug dosage combinations of Tacrolimus and Motolimod depicted by D-SPIN. The phase diagram shows the dominance of Motolimod-induced pathogen-response macrophage states over Tacrolimus-induced monocyte-like states.

## Key resources table

| REAGENT or RESOURCE | SOURCE | IDENTIFIER |
|---|---|---|
| Biological Samples | | |
| Human PBMCs from healthy donors | Hemacare | PB009C-50 |
| Chemicals | | |
| Small molecule drugs (Immunology and Inflammation library) | SelleckChem | L4100 |
| Small molecule drugs (FDA-approved drug library) | SelleckChem | L2000 |
| Reagents | | |
| RPMI-1640 | Gibco | 22400089 |
| Penicillin-Streptomycin (Pen/Strep) | Gibco | 15070063 |
| Fetal Bovine Serum (FBS) | Thermo Fisher | A3840001 |
| ImmunoCul Human CD3/CD28 T Cell Activator | Stemcell Technologies | 10971 |
| TrypLE Express Enzyme | Thermo Fisher | 12604021 |
| 10X Chromium Single Cell 3' Reagent Kits v3.1 | 10x Genomics | PN-1000121 |
| 10X Chromium Next GEM Single Cell 3' HT Reagent Kits v3.1 | 10x Genomics | 1000370 |
| 10X Chromium Cellplex Kit | 10x Genomics | 1000261 |
| MULTI-seq lipid-modified oligos | Gartner lab | [5] |
| Consumables | | |
| CellRepellent Surface Cell Culture Dishes | Greiner | 7000687/664970 |
| Cell-repellent treated 96-well plates | Greiner Bio-One | 07-000-640 |
| 96-well conical V-bottom plate | Thermo Fisher | 249935 |
| Equipment | | |
| Tecan evo 200 | Tecan | 30020020 |
| Software | | |
| Cell Ranger | 10x Genomics | Versions: 3.1.0; 5.0.1; 6.1.2 |
| MATLAB | MathWorks | Version: R2019a |

## Resource availability

### Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Matt Thomson (mthomson@caltech.edu).

## Experimental model and subject details

We thawed cryopreserved human peripheral blood mononuclear cells (PBMCs) from healthy donors (Hemacare, PB009C-50, Lot 20062904) in a 37 °C water bath and transferred these to a conical vial. Pre-warmed RPMI-1640 (Gibco 22400089) was then added at a 10:1 ratio relative to the original cell suspension to dilute the cryoprotectant DMSO. Cells were pelleted for 2 min at 300 rcf and then resuspended in pre-warmed RPMI-1640 + 1% Pen/strep (Gibco 15070063) + 10% Fetal Bovine Serum (Thermo Fisher A3840001) to a final cell density of $10^6$ cells/mL. Cells were plated into CELLSTAR CellRepellent Surface Cell Culture Dishes (Greiner 7000687/664970) and then rested in a tissue culture incubator at 37 °C, 5% CO2 for 16 hours before being exposed to small molecule drugs.

# 1 Method details

## 1.1 Overview of D-SPIN framework

The D-SPIN framework utilizes single-cell transcriptional profiling data from multiple conditions to construct a regulatory network model based on the gene programs identified, and the impacts of each perturbation to the network on the program level. The framework consists of three stages: gene program discovery, regulatory network model inference, and network analysis. Here, we first present a general overview and will delve into specific theories and implementations in subsequent sections.

Compared with existing methods of regulatory network inference, D-SPIN has three major advantages. First, D-SPIN integrates information from multiple perturbations into a unified regulatory network model, significantly improving the network reconstruction accuracy. Second, D-SPIN is generative and defines a complete probabilistic distribution of all transcriptional states with a set of minimal parameters of gene (program) interactions and perturbation bias, so that we can perform reasoning on the network model. Third, D-SPIN is highly scalable with the number of cells and conditions as the computation time scales linearly with cell number, and its algebraic structure enables parallelized inference.

For the gene program discovery step, D-SPIN requires gene programs to be a non-overlapping subset of genes, as shared genes between programs will cause confounding interactions in the network. D-SPIN is compatible with different methods for gene program extraction, such as unsupervised matrix decomposition, differential expression, gene set enrichment, and manual curation. Users can also pre-assign a set of interesting single genes or gene programs and let D-SPIN partition the remaining genes into gene programs. For each gene program (set of genes $\{g_i\}$), the expression level $l$ is determined by the weighted average of genes in the set that best approximates the expression of all genes in the set (Sec. 1.8). In our analysis, we use orthogonal non-negative matrix factorization (oNMF) to partition genes into gene programs (Sec. 1.8).

For the regulatory network inference step, D-SPIN applies a discretized spin-network modeling framework originating from statistical physics. The discretization can account for multiple different gene expression states while being resistant to counting noise in single-cell data. Further, discretization naturally generates multiple stable states and accommodates multiple cell types within a population and with a minimal interaction model, which can be interpreted as the steady state of a reaction system with nonlinear activations (Sec. 1.7). The number of possible states can be increased in a controlled way by changing the number of discretizations. The original Ising model from statistical physics uses two discretized states $\{-1, 1\}$, while we elected to use three discretized states $\{-1, 0, 1\}$ for both better interpretation and improved quantitative agreement between model and empirical cell state distribution by including two different levels of gene program expression (Sec. 1.2). In D-SPIN, each cell is represented as a state vector of gene programs $\boldsymbol{s} = [s_1, s_2, \cdots, s_M], \boldsymbol{s} \in \{-1, 0, 1\}^M$. Thus, D-SPIN serves as a probabilistic graphical model of the observed distribution of cell states that can be interpreted as a regulatory network.

As further explored in a related study [6], D-SPIN's improved network inference accuracy is achieved by integrating information across multiple conditions, so as to capture different modes of the underlying regulatory network. If a large proportion of experimental conditions have very similar cell state distributions, the network inference would be biased towards such dominating conditions. In such cases, we recommend only including a subset of conditions that have different cell state distributions from each other to balance the conditions used for inference (Sec. 2.4).

The regulatory network and response vectors inferred by D-SPIN can help define underlying principles of gene regulatory networks and identify classes of perturbations based on response signatures.

From the network perspective, biological networks commonly exhibit a modular organization where the network contains a set of strongly interacting subnetworks, each of which is associated with a specific biological function. We analyze D-SPIN regulatory networks with Leiden community detection to identify modular subnetworks (Sec. 2.11). Further, D-SPIN defines a probabilistic distribution of cell states only through the pairwise interaction between genes or gene programs, providing an interpretable framework for reasoning on the regulatory network. The reasoning framework also elucidates how each edge contributes to observed perturbation impact, as demonstrated in the edge sensitivity analysis (Sec. 2.7).

From the perturbation perspective, the response vectors provide compact representations of the impact of perturbations on the network, enabling the clustering and visualization of perturbation responses across multiple experimental batches with a simple batch effect correction scheme. Due to the linear form of $\boldsymbol{h}$ in the model, subtracting the response vector of the control sample calculates the up- or down-regulation of each gene program relative to the control. As the response vectors quantify the interactions between perturbations and the network that define the perturbation responses, joint graph clustering of network and perturbations can be applied to understand global perturbation action patterns and response strategies in the context of the regulatory network (Sec. 2.12).

## 1.2 Formulation of D-SPIN framework

The probabilistic graphical model we used in the formulation of D-SPIN originates from the Ising model in statistical physics, also called a spin glass, spin network, or Markov random field in different disciplines. The model defines a probability distribution on a discretized state vector $\boldsymbol{s} = [s_1, s_2, \cdots, s_M]$, $s_i \in \{-1, 0, 1\}$, where $M$ is the number of genes or gene programs. The probability distribution defined by the model is

$$P(\boldsymbol{s}) = \frac{1}{Z} \exp[-E(\boldsymbol{s}; \boldsymbol{J}, \boldsymbol{h}^{(n)})] \tag{1}$$

$$E(\boldsymbol{s}; \boldsymbol{J}, \boldsymbol{h}^{(n)}) = -\sum_{ij} J_{ij} s_i s_j - \sum_i h_i^{(n)} s_i \tag{2}$$

$$Z = \sum_{\boldsymbol{s}} \exp[-E(\boldsymbol{s}; \boldsymbol{J}, \boldsymbol{h}^{(n)})], \tag{3}$$

where $E$ is the energy function and $Z$ is a normalizing constant called partition function. States with lower energies have higher probabilities; therefore, positive interactions $J_{ij} > 0$ promote $s_i$ and $s_j$ being simultaneously on or off, and negative interactions promote the two variables being at different states. The bias vectors $h_i$ increase probability of the variable $s_i$ being on ($h_i > 0$) or off ($h_i < 0$).

D-SPIN assumes that across perturbation conditions, the interaction network $\boldsymbol{J}$ stays the same and the bias vector $\boldsymbol{h}^{(n)}$ is modulated by each perturbation condition $n$ to activate different modes in the network and to produce different cellular transcription state distributions. We train the model from data by gradient ascent of the log-likelihood function.

$$\log \mathcal{L}(\boldsymbol{J}, \boldsymbol{h}^{(n)}) = \sum_n^N \sum_{\boldsymbol{s}_{\text{Data}}^{(n)}} \log P(\boldsymbol{s}_{\text{Data}}^{(n)} | \boldsymbol{J}, \boldsymbol{h}^{(n)}) \tag{4}$$

We compute the gradient of the objective function as

$$\frac{\partial \log \mathcal{L}}{\partial J_{ij}} = \sum_n^N \sum_{\boldsymbol{s}_{\text{Data}}^{(n)}} \frac{\partial}{\partial J_{ij}} \left[ -E(\boldsymbol{s}_{\text{Data}}^{(n)}) - \log Z \right] \tag{5}$$

$$= \sum_n^N \sum_{\boldsymbol{s}_{\text{Data}}^{(n)}} \left\{ s_{i\text{Data}}^{(n)} s_{j\text{Data}}^{(n)} - \frac{1}{Z} \sum_{\boldsymbol{s}} s_i s_j \exp[-E(\boldsymbol{s})] \right\} \tag{6}$$

The first term is the sum $s_i s_j$ over all experimental samples, and the second term is a constant, which is the expectation of $s_i s_j$ of the current model, defined by the parameters $\boldsymbol{J}, \boldsymbol{h}$. We normalize the gradient by sample number to improve numerical stability under a given step size. With similar derivation for $\partial \log \mathcal{L} / \partial h_i$, the gradients of the objective function have the following form.

$$\frac{\partial \log \mathcal{L}}{\partial \boldsymbol{J}} = \frac{1}{N} \sum_n^N \left( \langle s_i s_j \rangle_{\text{Data}}^{(n)} - \langle s_i s_j \rangle_{\text{Model}}^{(n)} \right) \tag{7}$$

$$\frac{\partial \log \mathcal{L}}{\partial \boldsymbol{h}^{(n)}} = \langle s_i \rangle_{\text{Data}}^{(n)} - \langle s_i \rangle_{\text{Model}}^{(n)} \tag{8}$$

where $\langle \cdot \rangle_{\text{Data}}^{(n)}$ is the average over the data of $n$-th conditions, and $\langle \cdot \rangle_{\text{Model}}^{(n)}$ is the expectation on the distribution defined by current model parameters $\boldsymbol{J}$, $\boldsymbol{h}^{(n)}$. Therefore, the modeling learning is essentially matching the cross-correlation $\langle s_i s_j \rangle$ and mean $\langle s_i \rangle$ between the model and data.

The model has a few unique advantages:

1. From a statistical perspective, the spin network is the maximum entropy model given the mean and pairwise cross-correlation of data [7]. Entropy describes how a distribution is "spread out" over all possible states, and the principle of maximum entropy states that the best model describing a system is the model that generates data distributions with the maximum entropy while agreeing with relevant statistics of the data. Such a model has no extra assumptions about the structure of the system apart from the measured statistics.
2. From a computational perspective, the inference of the spin network, i.e., maximization of the log-likelihood function, is a concave problem where the only local maximum is the global maximum (Sec. 1.5). Therefore, optimization techniques to avoid traps of local minimums are not necessary. However, it is worth noting that the optimization is still a difficult NP-hard problem in the field of computational complexity [8], primarily due to the computational complexity of accurate gradient estimation and large condition numbers from model identifiability issues [9]. Inferring the model across perturbation conditions can mitigate these issues, and we computationally evaluated the curvature of the inference problem in our dataset (Sec. 1.6) and explored it further theoretically in another work [6].

3. From a physics perspective, the spin network is a Boltzmann distribution defined by the energy of each state. The model can be connected to the distribution of microscopic state in thermodynamics and dynamics of a microscopic system under thermal noise (Sec. 1.7)

There are a few potential choices of discretizing the expression level $\boldsymbol{l}$: the number of discretized states $m$ and the choice between $\{-\lfloor m/2 \rfloor, -\lfloor m/2 \rfloor + 1, \ldots, \lfloor m/2 \rfloor - 1, \lfloor m/2 \rfloor\}$ or $\{0, 1, \ldots, m\}$. We choose to discretize $\boldsymbol{l}$ into 3 states $\{-1, 0, 1\}$ due to the following reasons:

1. Larger $m$ increases computational complexity, and makes the model closer to a Gaussian distribution that has a single probability density maximum.
2. We find $m = 2$ is insufficient to characterize phenomena in our drug profiling experiments. We observed different levels of program activation instead of on-and-off switching. For example, both Glucocorticoid drugs and immune inhibitors activate the program P22 M2 macrophage, but with different expression levels.
3. The choice $\{-1, 0, 1\}$ is preferable because the self-interactions term $J_{ii}$ has more clear biological interpretations. Considering a single program only, self-activation $J_{ii} > 0$ is similar to a bi-stable switch produced by nonlinear activation, where $-1$ and 1 states are preferable. Similarly, self-inhibition $J_{ii} < 0$ is similar to negative feedback where the 0 state is preferable.
4. On the network inference benchmarking on synthetic networks, there is no significant difference between the choice of $\{-1, 0, 1\}$ and $\{0, 1, 2\}$.

## 1.3 Computational methods for inference in D-SPIN

Various computational methods have been proposed to solve the spin network inference problem [7], and we adapt three of the most accurate methods to the context of D-SPIN. Each method has its advantages and specific niches depending on the problem setting, such as network size, number of cells, and number of perturbation conditions.

**Exact maximum likelihood inference** For small networks, the probability distribution $P(\boldsymbol{s}|\boldsymbol{J}, \boldsymbol{h})$ can be explicitly computed, thus the exact gradient of $\boldsymbol{J}, \boldsymbol{h}$ can be computed using Eq. 7, Eq. 8. With the exact gradient, the optimization problem can be solved using gradient ascent or other optimizers such as Momentum or Adam.

As the gradient estimation requires enumerating all possible states, the computational complexity scales exponentially with the number of nodes. Though exact, the method is only applicable to small networks of around 10 nodes ($3^{10} \approx 6 \times 10^4$ states).

**Markov Chain Monte Carlo (MCMC) maximum likelihood inference** As the gradient only requires computing the mean and cross-correlation of the samples, we can approximate the complete distribution $P(\boldsymbol{s}|\boldsymbol{J}, \boldsymbol{h})$ by sampling an empirical distribution. Without evaluating the exact distribution, we can construct a Markov Chain between states whose stationary distribution is $P(\boldsymbol{s}|\boldsymbol{J}, \boldsymbol{h})$. Specifically, we utilize the Gibbs sampling scheme. Starting from a random initial state $\boldsymbol{s} = [s_1, s_2, \ldots, s_M]^\intercal$, at each step we randomly take an index $k$ in the $M$ nodes, and update the value of $s_k$ by its conditional distribution given all other nodes $P(s_k|s_1, \ldots, s_{k-1}, s_{k+1}, \ldots, s_M, \boldsymbol{J}, \boldsymbol{h})$. After a burn-in period of steps to allow the Markov Chain to equilibrate, the sequence of samples is an empirical distribution of $P(\boldsymbol{s}|\boldsymbol{J}, \boldsymbol{h})$ and can be used to estimate the gradient Eq. 7, Eq. 8.

In each sampling step, computing the marginal distribution is of complexity $\mathcal{O}(M)$. For accurate cross-correlation estimation, the required number of samples scales as $\mathcal{O}(M^2)$, without considering the burn-in period of the MCMC process. Therefore the overall computational complexity is at least $\mathcal{O}(M^3)$. The MCMC method applies to medium-sized networks up to $30 \sim 50$ nodes.

**Pseudolikelihood method** The major challenge involved in scaling inference to large networks is the partition function $Z$ in the distribution $P(\boldsymbol{s}|\boldsymbol{J}, \boldsymbol{h})$ which involves exponentially many terms with the network size $M$. An alternative approximation method called pseudolikelihood was developed originally for spatial statistics and adapted to spin network problems [10, 11, 7]. Pseudolikelihood methods approximate the distribution $P(\boldsymbol{s}|\boldsymbol{J}, \boldsymbol{h})$ as the product of the conditional distribution of each individual variable, and provide exact inference in the limit of an infinite number of samples.

Specifically, we denote $\boldsymbol{s}_{\setminus k} = [s_1, \ldots, s_{k-1}, s_{k+1}, \ldots, s_M]^\intercal$ as the state $\boldsymbol{s}$ except $s_k$. The pseudolikelihood

function for the inference problem of a single condition is

$$\text{Pseudo}P(\boldsymbol{s}|\boldsymbol{J},\boldsymbol{h}) = \prod_k^M P(s_k|\boldsymbol{s}_{\setminus k},\boldsymbol{J},\boldsymbol{h}) = \prod_k^M \frac{\exp E(s_k,\boldsymbol{s}_{\setminus k},\boldsymbol{J},\boldsymbol{h})}{\sum_{r\in\{-1,0,1\}}\exp E(s_k=r,\boldsymbol{s}_{\setminus k},\boldsymbol{J},\boldsymbol{h})} \tag{9}$$

$$= \prod_k^M \frac{\exp(s_k\Theta_k + s_k^2 J_{kk})}{\exp(\Theta_k + J_{kk}) + 1 + \exp(-\Theta_k + J_{kk})}\,, \tag{10}$$

where $\Theta_k = h_k + \sum_{j\neq k} J_{jk}s_j$ is the effective field conditioned on all other nodes $\boldsymbol{s}_{\setminus k}$. The pseudolikelihood function decouples the mutual dependence between nodes, thus removing the exponentially complex partition function $Z$. As a cost, the pseudolikelihood function in general does not sum to 1, and thus is not a distribution; this is where the "pseudo" name comes from. The gradient for the log pseudolikelihood objective function is

$$\frac{\partial\mathcal{L}_{\text{Pseudo}}}{\partial J_{ij,i\neq j}} = \frac{\partial}{\partial J_{ij}}\sum_k^M \left(s_k\Theta_k + s_k^2 J_{kk} - \log\left(\exp(\Theta_k + J_{kk}) + 1 + \exp(-\Theta_k + J_{kk})\right)\right)$$

$$= 2s_is_j - \sum_{k\in\{i,j\}} \frac{\exp(\Theta_k + J_{kk}) - \exp(-\Theta_k + J_{kk})}{\exp(\Theta_k + J_{kk}) + 1 + \exp(-\Theta_k + J_{kk})} \tag{11}$$

$$\frac{\partial\mathcal{L}_{\text{Pseudo}}}{\partial J_{ii}} = s_i^2 - \sum_{k=i} \frac{\exp(\Theta_k + J_{kk}) + \exp(-\Theta_k + J_{kk})}{\exp(\Theta_k + J_{kk}) + 1 + \exp(-\Theta_k + J_{kk})} \tag{12}$$

$$\frac{\partial\mathcal{L}_{\text{Pseudo}}}{\partial h_i} = s_i - \sum_{k=i} \frac{\exp(\Theta_k + J_{kk}) - \exp(-\Theta_k + J_{kk})}{\exp(\Theta_k + J_{kk}) + 1 + \exp(-\Theta_k + J_{kk})}\,. \tag{13}$$

The computational complexity of the gradient computation scales with $\mathcal{O}(M^2)$. Also, note that the effective field $\Theta_k$ depends on each cell state $\boldsymbol{s}$, so the gradient computational scales with the total number of cells. Practically, the pseudolikelihood method is highly scalable and even applies to networks of thousands of nodes. The approximation of pseudolikelihood is more accurate when the number of samples is high. However, typically the number of observed samples is far lower than the total number of possible states ($3^M$). Therefore exact maximum likelihood and MCMC maximum likelihood are preferred when they are computationally feasible.

Furthermore, the form of pseudolikelihood is closely related to regression models, enabling assigning directionality to the inferred network. The distribution of a single node conditioned on all other nodes

$$P(s_k|\boldsymbol{s}_{\setminus k},\boldsymbol{J},\boldsymbol{h}) = \frac{\exp(s_k\Theta_k + s_k^2 J_{kk})}{\exp(\Theta_k + J_{kk}) + 1 + \exp(-\Theta_k + J_{kk})} \tag{14}$$

can be interpreted as a regression problem, where we predict the state of the dependent variable $s_k$ with other variables are predictors. The interactions $J_{ij}$ are the coefficients of the regression problem. If gene A predicts gene B better than B predicts A, this suggests a regulation direction of A to B. To compute the directional network, the gradient estimation Eq. 11 can be simply replaced with an asymmetric version of

$$\frac{\partial\mathcal{L}_{\text{Pseudo}}}{\partial J_{ij,i\neq j}} = s_is_j - \sum_{k=i} \frac{\exp(\Theta_k + J_{kk}) - \exp(-\Theta_k + J_{kk})}{\exp(\Theta_k + J_{kk}) + 1 + \exp(-\Theta_k + J_{kk})} \tag{15}$$

Even though D-SPIN is capable of inferring a directed network, we focus our analysis on undirected networks because directed networks cannot define a stationary distribution on the data when time information is not supplied. In the context of probabilistic graphical models, directed models are always constrained to be acyclic, i.e. with no cycle. Such a constraint is reasonable in the field of causal inference, where the circularity of causal relations is rare. However, in cellular regulatory networks, feedback loops are prevalent to maintain homeostasis or signal amplification. For example, regulatory network models of hematopoietic stem cell differentiation contain several loops, such as Pu1-Gata1-Gata2-Pu1 [12].

The acyclic constraint of directed probabilistic graphical models is fundamental, as cycles in the conditional dependence between variables will produce inconsistent distributions. As mentioned in the seminal work of Judea Pearl on causal inference [13, 14], even three cyclic dependent variables "will normally lead to inconsistencies", and the acyclic constraint can ensure consistency of the distributions. Constructing distributions on even two mutually dependent variables requires nontrivial constraints [15]. Another approach to include directionality in regulatory network models is including explicit time dependence, which also has limited application due to the lack of experimental approaches to measure dynamical evolution of transcriptome profiles. Therefore, in the context of probabilistic models of regulatory networks, undirected networks are a more appropriate choice.

## 1.4 Regularization of the inference

In statistical inference, it is common to leverage prior knowledge about the potential form of the solution, a process known as regularization. Regularization nudges the solution towards a preferred direction, which can also be interpreted as assigning a prior distribution of the model parameters in the framework of Bayesian inference. For example, $\ell_1$ (Lasso) regularization promotes sparsity of the solution, while $\ell_2$ (Ridge) regularization promotes solutions with smaller magnitudes.

Specifically in the D-SPIN framework, we sometimes have prior knowledge of perturbation action from the experimental design. For example, in single-cell profiling of gene knockdown or activation, the response vector relative to control should have a strong inhibition/activation at the target gene. Suppose we have an estimation of the relative action of all the perturbations as $\boldsymbol{h}_r^{(n)}$. Then we can infer shared unperturbed single gene (program) activity $\boldsymbol{h}_0$ and penalize the difference between $\boldsymbol{h}^{(n)} - \boldsymbol{h}_0$ and $\boldsymbol{h}_r^{(n)}$ by $\ell_2$ norm. In this case, the objective function becomes

$$\log \mathcal{L}(\boldsymbol{J}, \boldsymbol{h}^{(n)}) = \sum_n^N \sum_{\boldsymbol{s}_{\text{Data}}^{(n)}} \log P(\boldsymbol{s}_{\text{Data}}^{(n)} | \boldsymbol{J}, \boldsymbol{h}^{(n)}, \boldsymbol{h}_0) - \lambda \sum_n^N (\boldsymbol{h}^{(n)} - \boldsymbol{h}_0 - \boldsymbol{h}_r^{(n)})^2 \,, \tag{16}$$

where $\lambda$ is the strength of the regularization, representing the uncertainty of the prior knowledge of the relative response estimation. For this objective function, the gradient of $\boldsymbol{J}$ stays the same and

$$\frac{\partial \log \mathcal{L}}{\partial \boldsymbol{h}_0} = \frac{1}{N} \sum_n^N \left( \langle s_i \rangle_{\text{Data}}^{(n)} - \langle s_i \rangle_{\text{Model}}^{(n)} \right) \tag{17}$$

$$\frac{\partial \log \mathcal{L}}{\partial \boldsymbol{h}^{(n)}} = \langle s_i \rangle_{\text{Data}}^{(n)} - \langle s_i \rangle_{\text{Model}}^{(n)} - 2\lambda (\boldsymbol{h}^{(n)} - \boldsymbol{h}_0 - \boldsymbol{h}_r^{(n)}) \tag{18}$$

## 1.5 Proof of convexity of the inference problem

We prove the multi-condition/multi-$\boldsymbol{h}$ inference problem by D-SPIN is a convex optimization problem. Specifically, the log-likelihood objective function is concave. Therefore the only local optimum is the global optimum, and the solution to the optimization problem is unique.

For a single condition, D-SPIN formulation is the same as the inverse Ising problem, whose log-likelihood function is concave, and strictly concave if all coupling and fields are finite. We follow the proof from a review [7]. We denote $\boldsymbol{\lambda} = \{\boldsymbol{J}, \boldsymbol{h}\}$, and $Q_k(\boldsymbol{s}) = \{s_i s_j, s_i\}$ for the model parameters and the log-likelihood can be written as

$$\log \mathcal{L} = \sum_k \lambda_k Q_k - \log Z(\boldsymbol{\lambda}) \,. \tag{19}$$

The second derivative of the log-likelihood function follows

$$\begin{aligned} -\frac{\partial^2 \log \mathcal{L}}{\partial \lambda_i \lambda_j} &= \frac{\partial^2 \log \sum_{\boldsymbol{s}} \exp \sum_k \lambda_k Q_k}{\partial \lambda_i \lambda_j} = \frac{\partial}{\partial \lambda_j} \frac{\sum_{\boldsymbol{s}} Q_i \exp \sum_k \lambda_k Q_k}{\sum_{\boldsymbol{s}} \exp \sum_k \lambda_k Q_k} \\ &= \frac{\sum_{\boldsymbol{s}} Q_i Q_j \exp \sum_k \lambda_k Q_k}{\sum_{\boldsymbol{s}} \exp \sum_k \lambda_k Q_k} - \frac{\left(\sum_{\boldsymbol{s}} Q_i \exp \sum_k \lambda_k Q_k\right) \left(\sum_{\boldsymbol{s}} Q_j \exp \sum_k \lambda_k Q_k\right)}{\left(\sum_{\boldsymbol{s}} \exp \sum_k \lambda_k Q_k\right)^2} \\ &= \langle Q_i Q_j \rangle - \langle Q_i \rangle \langle Q_j \rangle \,. \end{aligned} \tag{20}$$

The negative matrix of second derivatives $H = -\frac{\partial^2}{\partial \boldsymbol{\lambda}^2} \log \mathcal{L}$ is positive semidefinite because for any vector $\boldsymbol{x}$,

$$\boldsymbol{x}^\intercal H \boldsymbol{x} = \sum_{ij} (\langle Q_i Q_j \rangle - \langle Q_i \rangle \langle Q_j \rangle) x_i x_j = \left\langle \left[ \sum_k (x_k Q_k - \langle x_k Q_k \rangle) \right]^2 \right\rangle \geq 0 \,. \tag{21}$$

Further, the matrix is positive definite if no observable $Q_k$ has vanishing fluctuations, which holds when all the couplings and fields are finite. Therefore the log-likelihood function is strictly concave with finite couplings and fields, and the maximum likelihood estimation is a convex optimization problem.

For the formulation of D-SPIN of multiple conditions, we prove the case with two conditions and the proof of more conditions naturally follows. We start with proving a lemma:

Lemma: Let $A1, A2, B1, B2, C1, C2$ be matrices such that the block matrices $\begin{bmatrix} A1 & B1 \\ B1^\intercal & C1 \end{bmatrix}$ and $\begin{bmatrix} A2 & B2 \\ B2^\intercal & C2 \end{bmatrix}$ are both positive (semi)definite. Then the block matrix

$$\begin{bmatrix} A1 + A2 & B1 & B2 \\ B1^\intercal & C1 & 0 \\ B2^\intercal & 0 & C2 \end{bmatrix} \tag{22}$$

is also positive (semi)definite.

Proof: For any test vector $\boldsymbol{x}$, we denote $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \end{bmatrix}$, where each component is compatible with the shape of $A_1 + A_2, C_1$ and $C_2$, then we have

$$\boldsymbol{x}^\intercal \begin{bmatrix} A1 + A2 & B1 & B2 \\ B1^\intercal & C1 & 0 \\ B2^\intercal & 0 & C2 \end{bmatrix} \boldsymbol{x} = \boldsymbol{x}_1^\intercal (A_1 + A_2)\boldsymbol{x}_1 + 2\boldsymbol{x}_1^\intercal B_1 \boldsymbol{x}_2 + 2\boldsymbol{x}_1^\intercal B_2 \boldsymbol{x}_3 + \boldsymbol{x}_2^\intercal C_1 \boldsymbol{x}_2 + \boldsymbol{x}_3^\intercal C_2 \boldsymbol{x}_3 \tag{23}$$

$$= \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix}^\intercal \begin{bmatrix} A1 & B1 \\ B1^\intercal & C1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_3 \end{bmatrix}^\intercal \begin{bmatrix} A2 & B2 \\ B2^\intercal & C2 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_3 \end{bmatrix} (\geq) > 0 \tag{24}$$

as both matrices are positive (semi)definite.

Similarly, we denote the parameter set $\boldsymbol{\lambda} = \{\boldsymbol{J}, \boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}\}$, and the log-likelihood function of the two conditions are $\log \mathcal{L}^{(1)}$, $\log \mathcal{L}^{(2)}$, and the second order derivative of the objective function follows

$$\frac{\partial(\log \mathcal{L}^{(1)} + \log \mathcal{L}^{(2)})}{\partial \lambda_i \lambda_j} = \begin{bmatrix} \frac{\partial \log \mathcal{L}^{(1)}}{\partial J_i J_j} + \frac{\partial \log \mathcal{L}^{(2)}}{\partial J_i J_j} & \frac{\partial \log \mathcal{L}^{(1)}}{\partial J_i h_j^{(1)}} & \frac{\partial \log \mathcal{L}^{(2)}}{\partial J_i h_j^{(2)}} \\ \frac{\partial \log \mathcal{L}^{(1)}}{\partial J_i h_j^{(1)}}^\intercal & \frac{\partial \log \mathcal{L}^{(1)}}{\partial h_i h_j} & 0 \\ \frac{\partial \log \mathcal{L}^{(2)}}{\partial J_i h_j^{(2)}}^\intercal & 0 & \frac{\partial \log \mathcal{L}^{(2)}}{\partial h_i h_j} \end{bmatrix} \tag{25}$$

which is the rearrangement of the block matrices of the second-order derivative of the inverse Ising problem.

According to the lemma, the negative of the second-order derivative matrix is positive semidefinite, and positive definite if all couplings and fields are finite. This concludes our proof that the formulation of D-SPIN is a convex optimization problem.

## 1.6 Quantification of the inference uncertainty

Even though the optimization problem of D-SPIN inference has a unique solution, there is a possibility that there are other sub-optimal solutions that are distinct from the true solution but have similar values of the objective function. These alternative solutions are phenomenologically referred as "flat valleys" of the optimization landscape, or "sloppy directions" of models. This flatness is quantified by Fisher information, which is a metric describing the curvature in the space of a parametric family of probability distributions. Large Fisher information indicates that small changes in the model parameter cause great changes in the distribution, while small Fisher information indicates that the distribution is insensitive to the parameter changes.

The Fisher information and uncertainty of the inference are connected by the Cramér-Rao bound, that for any unbiased estimator, the variance is lower bounded by the inverse of the Fisher information. Specifically, the bound is achieved by efficient estimators, and the maximum likelihood estimator is asymptotically efficient. Therefore, estimating the Fisher information matrix of the inference problem provides an estimation of the variance of the inferred network.

Specifically, using the same notation $\boldsymbol{\lambda} = \{\boldsymbol{J}, \boldsymbol{h}\}$, and $Q_k(\boldsymbol{s}) = \{s_i s_j, s_i\}$ as the previous section the Fisher information matrix of a single condition can be computed by definition

$$\mathcal{I}(\boldsymbol{\lambda}) = -\mathbb{E}\left[\frac{\partial^2}{\partial \boldsymbol{\lambda}^2} \log P(\boldsymbol{s}|\boldsymbol{\lambda})\right] = -\mathbb{E}\left[\frac{\partial^2}{\partial \boldsymbol{\lambda}^2}\left(\sum_k \lambda_k Q_k - \log Z(\boldsymbol{\lambda})\right)\right] \tag{26}$$

$$= \langle Q_i Q_k \rangle - \langle Q_i \rangle \langle Q_k \rangle \,, \tag{27}$$

which has exactly the same form as Eq. 20 as the first linear term $\sum_k \lambda_k Q_k$ vanishes under second order derivative. The bracket $\langle \cdot \rangle$ is average over $P(\boldsymbol{s}|\boldsymbol{J}, \boldsymbol{h})$, which is not directly available but can be estimated by the empirical distribution defined by samples. In practice, the Fisher information of a specific condition is estimated by

$$\mathcal{I}(J_{ij}, J_{kl}) = \langle s_i s_j s_k s_l \rangle_{\text{Data}} - \langle s_i s_j \rangle_{\text{Data}} \langle s_k s_l \rangle_{\text{Data}} \tag{28}$$

$$\mathcal{I}(J_{ij}, h_k) = \langle s_i s_j s_k \rangle_{\text{Data}} - \langle s_i s_j \rangle_{\text{Data}} \langle s_k \rangle_{\text{Data}} \tag{29}$$

$$\mathcal{I}(h_i, h_k) = \langle s_i s_k \rangle_{\text{Data}} - \langle s_i \rangle_{\text{Data}} \langle s_k \rangle_{\text{Data}} \,, \tag{30}$$

where the bracket $\langle \cdot \rangle_{\text{Data}}$ is average over all states in the data.

In the case of multiple samples, the overall Fisher information of the Fisher information has a similar form as Eq. 25. We also only write for two conditions for simplicity of notation.

$$\mathcal{I} = \begin{bmatrix} \mathcal{I}^{(1)}_{(\boldsymbol{J},\boldsymbol{J})} + \mathcal{I}^{(2)}_{(\boldsymbol{J},\boldsymbol{J})} & \mathcal{I}^{(1)}_{(\boldsymbol{J},\boldsymbol{h})} & \mathcal{I}^{(2)}_{(\boldsymbol{J},\boldsymbol{h})} \\ \mathcal{I}^{(1)\intercal}_{(\boldsymbol{J},\boldsymbol{h})} & \mathcal{I}^{(1)}_{(\boldsymbol{h},\boldsymbol{h})} & 0 \\ \mathcal{I}^{(2)\intercal}_{(\boldsymbol{J},\boldsymbol{h})} & 0 & \mathcal{I}^{(2)}_{(\boldsymbol{h},\boldsymbol{h})} \end{bmatrix} \tag{31}$$

As we have proved $\mathcal{I}$ is positive definite when all couplings and fields are finite, $\mathcal{I}_{(\boldsymbol{J},\boldsymbol{J})}$ and $\mathcal{I}_{(\boldsymbol{h},\boldsymbol{h})}$ are all invertible. According to the Schur complement of the block matrix, we have

$$\begin{bmatrix} A & B \\ B^\intercal & C \end{bmatrix}^{-1} = \begin{bmatrix} (A - BC^{-1}B^\intercal)^{-1} & -(A - BC^{-1}B^\intercal)^{-1}BC^{-1} \\ -C^{-1}B^\intercal(A - BC^{-1}B^\intercal)^{-1} & C^{-1} + C^{-1}B^\intercal(A - BC^{-1}B^\intercal)^{-1}BC^{-1} \end{bmatrix} \tag{32}$$

Therefore the Cramé-Rao bound of the network estimation for $N$ conditions is

$$\mathcal{I}^{-1}_{(\boldsymbol{J},\boldsymbol{J})} = \left[ \sum_n^N C_n \left( \mathcal{I}^{(n)}_{(\boldsymbol{J},\boldsymbol{J})} - \mathcal{I}^{(n)}_{(\boldsymbol{J},\boldsymbol{h})} \mathcal{I}^{(n)}_{(\boldsymbol{h},\boldsymbol{h})} \mathcal{I}^{(n)\intercal}_{(\boldsymbol{J},\boldsymbol{h})} \right) \right]^{-1}, \tag{33}$$

where $C_n$ is the number of cells in the $n$-th condition. Specifically for each network edge $J_{ij}$, the diagonal term of the Cramé-Rao bound $\mathcal{I}^{-1}_{(J_{ij},J_{ij})}$ quantifies the variance of the inference uncertainty of the specific edge as presented in Figures S3G and S5E.

The inverse of Fisher information $\mathcal{I}^{-1}_{(\boldsymbol{J},\boldsymbol{J})}$ is the covariance matrix of the inferred network. Further, according to the Bernstein-von Mises theorem, the posterior distribution of the inferred network will converge to a normal distribution

$$P(\boldsymbol{J}|\{\boldsymbol{s}\}^{(1)}, \ldots, \{\boldsymbol{s}\}^{(N)}) = \mathcal{N}(\boldsymbol{J}_0, \mathcal{I}^{-1}_{(\boldsymbol{J},\boldsymbol{J})}), \tag{34}$$

where $\boldsymbol{J}_0$ is the result of inference. Therefore, we can sample from the posterior for alternative networks to validate the robustness of the inferred regulatory network as presented in Figures S3G and S5E.

## 1.7 Correspondence between D-SPIN and biochemical reaction systems

D-SPIN is closely connected to dynamical models of biochemical reactions as an approximation of the steady state of a chemical reaction system with saturating non-linear activation functions. For a biochemical reaction system with state $x \in \mathbb{R}^M$, the general dynamics can be written as

$$d\mathbf{X}_t = \boldsymbol{\mu}(\mathbf{X}_t, t)\, dt + \boldsymbol{\sigma}(\mathbf{X}_t, t)\, d\mathbf{W}_t \tag{35}$$

where $\boldsymbol{\mu}(\mathbf{X}_t, t)$ are the reactions and $\boldsymbol{\sigma}(\mathbf{X}_t, t)\, d\mathbf{W}_t$ is the stochastic noise. If the stochastic noise can be assumed a constant $\sigma$, then the state distribution $P(\boldsymbol{x})$ follows the Fokker-Planck equation

$$\frac{\partial P(\mathbf{x}, t)}{\partial t} = -\sum_{i=1}^N \frac{\partial}{\partial x_i} \left[\mu_i(\mathbf{x}, t) P(\mathbf{x}, t)\right] + \sum_{i=1}^N \sum_{j=1}^N \frac{\partial^2}{\partial x_i\, \partial x_j} \left[\frac{\sigma^2}{2} P(\mathbf{x}, t)\right]. \tag{36}$$

Specifically, if the reaction term can be written as the derivative of a potential function $\mu_i(\boldsymbol{x}) = -\partial V(\boldsymbol{x})/\partial x_i$, the equation has a Boltzmann distribution as the steady state solution for $\partial P(\boldsymbol{x}, t)/\partial t = 0$

$$P(\boldsymbol{x}) = \frac{1}{Z} \exp \frac{-2V(\boldsymbol{x})}{\sigma^2} \tag{37}$$

where $Z$ is the normalization constant. A typical example is a linear expansion near the stable point of a dynamical system, where

$$\mu_i(\boldsymbol{x}) = \sum_j a_{ij} x_j + b_i = -\frac{\partial}{\partial x_i}(-\sum_{ij} a_{ij} x_i x_j - \sum_i b_i x_i). \tag{38}$$

This solution is a Gaussian distribution near the stable point [16, 17]. Such models only have a single minimum in the convex potential function, and are therefore not suitable for characterizing cell populations with various stable cell types and cell states.

The potential function, i.e. energy function Eq. 2 in D-SPIN, is in a similar form to the Gaussian distribution as a second-order polynomial of variables. However, the energy in D-SPIN is a function of discretized state variables, which can be viewed as the discrete limit of composing the continuous state $x_i$ with a saturating activation function $s_i = \phi(x_i)$, for example, sigmoid function for a two-state model. Given the potential function, the reaction term in the corresponding dynamical model is

$$\mu_i(\boldsymbol{x}) = -\frac{\partial}{\partial x_i}(-\sum_{ij} J_{ij} s_i s_j - \sum_i h_i s_i) \tag{39}$$

$$= \phi(x_i)'(\sum_j J_{ij} \phi(x_j) + h_i) \tag{40}$$

The first term $\phi(x_i)'$ is the derivative of the activation function, which can be approximated by a constant in the transition regime of the activation function, and close to 0 at the saturating regime of the activation function. The corresponding interpretation is that during the transition between states, the regulation received by component $i$ is a linear function of the activation $\phi(x_j)$ of all other components $j$. The nonlinearity is conceptually similar to the Hill function used in models of genetic circuits and allows the existence of multiple stable states controlled by the regulatory networks in the cell.

## 1.8 Gene program discovery by oNMF

We use orthogonal nonnegative matrix factorization (oNMF) for gene program discovery because oNMF generates a set of programs that are mathematically constrained to provide a high-accuracy representation of transcriptional states in the data but with no overlap between gene programs. Compared to typical matrix factorization methods like principal components analysis (PCA), oNMF applies two constraints to the gene programs: non-negative weights and orthogonality. We used oNMF based on the following considerations:

1. Linearity: To ensure the interpretability of the model, each gene program is a linear combination of single genes
2. Non-negativity: The non-negative constraint avoids the ambiguity of interpreting negative weights, ensuring the programs are a set of co-expressed genes. Specifically in the context of regulatory network models, negative components in methods like principal components analysis (PCA) would complicate the interpretation of activation and inhibition interaction between gene programs.
3. Orthogonality: The orthogonality constraint makes it easier to interpret the data and aids model construction by forcing representations to be non-overlapping. Each gene program can be interpreted as a set of biological functions and each cell is represented by a collection of biological function activities. Without orthogonality, each cell is likely to be represented by a single program containing all genes expressed by the same cell type. Further, in the context of D-SPIN, shared genes between gene programs would cause confounding interactions between the programs. Therefore each gene program should be completely independent: in other words, orthogonal.

The formulation of oNMF is, given a non-negative gene matrix $X \in \mathbb{R}^{n_\text{cell} \times n_\text{gene}}$ and a program number $K$, oNMF solves the following optimization problem.

$$\begin{aligned} \underset{W,H}{\text{minimize}} \quad & \|X - WH\|_F^2 \\ \text{subject to} \quad & HH^\intercal = I, \quad W_{ij} \geq 0, \quad H_{ij} \geq 0 \end{aligned} \tag{41}$$

where $H \in \mathbb{R}^{K \times n_\text{gene}}$ is the gene program representation, $W \in \mathbb{R}^{n_\text{cell} \times K}$ is the cell state represented on the gene programs and $\|\cdot\|_F^2$ is the matrix Frobenius norm. We implemented the iterative algorithm proposed in this study [18], with $W$ and $H$ randomly initialized by uniform distribution on $[0, 1]$. The random initialization of $H$ is orthogonalized using singular value decomposition (SVD) and takes absolute value before the start of iteration; we use 500 iterations in the computation by default.

According to the mathematical formulation of oNMF, the influence of each gene and cell state on the objective function is determined by the norm of the reconstruction error. To improve the discovery of gene programs, we utilized the following strategies to balance the contributions from each gene and cell state:

1. Zero-inflated Poisson distribution gene CV filtering (Sec. 2.1).
   We noted that genes with high expression levels tended to have greater variance than typically expected under Poisson distribution CV filtering. Consequently, these genes often encode proteins involved in multiple housekeeping programs in the oNMF. Using a zero-inflated Poisson distribution for CV filtering, we accounted for the higher variance induced by dropouts, enabling better gene selection. We also removed genes that are expressed in a tiny fraction of cells in gene filtering, as genes with close to 0 expressions are not informative and risk introducing extra noise.
2. Gene normalization by standard deviation.
   Genes with high variance contribute more to the reconstruction error and can thus dominate the oNMF gene program discovery. To mitigate this, we divided the expression of each gene by its standard deviation across cells, ensuring that each gene contributed equally. The expression of genes with only a few non-zero entries may be disproportionately large after the scaling. Ideally, such genes should be filtered out during gene filtering, but they can also be removed by setting a cap of the scaling factor, for example, no more than 5 times increase.
3. Cell state balancing by cell clusters.
   The gene program discovery will be dominated by reconstructing a specific cell state if that state constitutes a large proportion of the cell population. To address this, we used class-balance strategies frequently applied in machine learning practices. Specifically, we balanced different cell states by sub-sampling cells based on cell clusters obtained by clustering or gene markers. We employed two different schemes: (1) Equal-sample balancing: to identify gene programs that are only expressed in a small fraction of the cell population, we sampled an equal number of cells from each cell cluster. (2) Square-root balancing: in machine learning, we found that taking samples from each data category proportional to the square root of the category size was effective. Similarly, we took samples from each cell cluster proportional to the square root of the cell number in that cluster.

The exact solution of non-negative matrix factorization is an NP-hard problem [19]. But there are heuristic approximations of oNMF by iterative matrix update, and the solution varies with random initialization. Therefore, we run oNMF with different random seeds and compute a consensus gene program decomposition. For example, we take each row of $H$ across different random seed repeats, and perform K-means clustering to compute the consensus composition of each gene program. The gene programs can be annotated with a combination of bioinformatics databases including DAVID, Enrichr, and String-db [1, 2, 20], and also manual lookup.

## 1.9 Evaluating gene program expression and discretization

A gene program is defined as a set of genes that co-express across the conditions in single-cell transcriptional profiling. Gene programs can either be identified through unsupervised learning techniques, such as oNMF (Sec. 1.8), or prior biological knowledge. In the D-SPIN framework, the expression level of gene program $k$ of cell $i$ is described by the variable $S_{ik} \in \{-1, 0, 1\}$. To transform the expression matrix of genes in the program into the discretized value for each cell, two steps are performed consecutively: computing a continuous weighted average of expression level $L_{ik}$, and discretizing into three levels.

**Weighted average** To reduce noise and to synthesize the expression of all genes in the program, the continuous expression level across all cells $\boldsymbol{l}_k = L_{\cdot,k}$ should be a weighted average of genes in the program. Denote the gene matrix as $G$, the index of genes in program $k$ as $\{g^{(k)}\}$ (which is the index of nonzero elements in $k$-th row of $H$ in the context of oNMF), the gene expression of cell $i$ on the program is $G_{i,\{g^{(k)}\}}$. The optimal weight $w_{g^{(k)}}$ and the expression level $L_{ik}$ that best characterize the gene expression as the solution to the optimization problem

$$\min_{w_{g^{(k)}}, L_{ik}} \quad \sum_i \sum_{\{g^{(k)}\}} (G_{i,\{g^{(k)}\}} - w_{g^{(k)}} L_{ik})^2 \tag{42}$$

which is summing over the error of reconstructing the gene expression with $\boldsymbol{l}_k$ and weight $w_{g^{(k)}}$ of every gene in the program. This is essentially a 1-component non-negative matrix factorization (NMF) problem and has mature implementations in Python scikit-learn packages.

**Discretization** Various methods have been empirically used to partition continuous expression into discrete levels, such as using percentiles or standard deviations. Discretization with K-means was found to perform well [21] and have a clear interpretation of minimizing the variance inside each expression level category. The 3-state K-means minimize the following objective function for each gene program $k$ across all cells indexed by $i$

$$\min_{S_{ik}} \quad \sum_{r \in \{-1,0,1\}} \sum_{S_{ik}=r} \|L_{ik} - \text{Mean}\{L_{ik}|S_{ik}=r\}\|^2 = \sum_{r \in \{-1,0,1\}} \#\{S_{ik}=r\}\text{Var}\{L_{ik}|S_{ik}=r\}\,, \tag{43}$$

which is assigning $-1, 0, 1$ to each of $S_{ik}$ so to minimize the variance of $L_{ik}$ inside each group of cells classified by $S_{ik}$. The K-means clustering also has mature implementations in Python scikit-learn packages.

## 1.10 Selection of gene program number in oNMF

The selection of the number of gene programs is a trade-off between model expressive power, computational complexity, and model interpretability. The optimal choice of program number typically hinges on the specific requirements of the application scenario. In the context of single-cell transcriptional profiling, the number of gene programs is typically set to $10 \sim 40$, depending on the desired resolution of the gene matrix decomposition. To aid in this decision, various model selection criteria have been proposed to facilitate this choice, among which the Bayesian information criterion (BIC) and elbow method are widely used examples.

BIC is derived from maximizing the model evidence $P(\text{Data}|\text{Model})$, where a penalty factor of the model dimension arises when integrating the parameter prior distribution over the whole parameter space [22]. Specifically, BIC is defined by maximized model log-likelihood $\hat{\mathcal{L}}$ with a linear penalty term of model dimension.

$$\text{BIC} = k \log n - 2 \log \hat{\mathcal{L}}\,, \tag{44}$$

where $k$ is the model dimension and $n$ is the number of samples. Models with lower BIC are more preferred.

As oNMF is not a probabilistic model by definition, BIC is not directly applicable. However, we can model the residue of the gene expression matrix after the oNMF fitting. In general, if the model error can be described as independent and identically distributed normal distribution variables,

$$\log \hat{\mathcal{L}} = \log \prod_i^n \frac{1}{\sqrt{2\pi}\sigma} \exp \frac{(x_i - \hat{x}_i)^2}{2\sigma^2} = -n \log \sigma - n \log \sqrt{2\pi} + \sum_i^n \frac{(x_i - \hat{x}_i)^2}{2\sigma^2}\,, \tag{45}$$

where $x_i$ and $\hat{x}_i$ are data and model fitting, $\sigma$ is the standard deviation of the error. As $\sigma$ for a model is generally unknown and estimated from data, we have $\sigma^2 = \sum_i^n (x_i - \hat{x}_i)^2/n$. Therefore the BIC is

$$\text{BIC} = k \log n - n \log \sigma^2 + C = k \log n - n \log(RSS/n) + C\,, \tag{46}$$

where $C$ is a constant independent of model choice and $RSS = \sum_i^n (x - \hat{x})^2$ is the residual sum of squares. Motivated by this form, in oNMF we can use the matrix Frobenius norm of the fitting residual as $RSS$, number of cells as $n$, and number of programs as $k$ to compute BIC to help to decide the number of gene programs.

Aside from statistical criteria like the BIC, the elbow method provides a heuristic approach to estimate the number of gene programs required in the oNMF. This method involves plotting a relevant cost function or objective function against the number of gene programs and looking for a point in the plot where the rate of changes drastically alters, resembling an "elbow".

Gene matrices of single-cell profiling are known to be especially noisy; therefore traditional metrics such as model-explained variance may not be effective in revealing the elbow point. Instead, we use the number of genes that significantly correlate with corresponding gene programs as a more robust objective function. To consider the effect of noise, we can evaluate the pairwise gene-gene correlation as a reference distribution, which has a standard deviation in the order of 0.05 in our datasets. The correlation threshold between genes and programs can be set as $3\sigma$ or $4\sigma$ of the gene-gene correlations. Additionally, various methods (including MAGIC) have been proposed to denoise the gene expression matrix [3]. After denoising, both gene-gene correlation and gene-program correlation drastically increase, allowing us to use a high correlation threshold, such as 0.5. These refined approaches would facilitate the decision of gene program numbers in D-SPIN.

## 1.11 High-throughput drug perturbation experiments on PBMCs

All high-throughput drug perturbation experiments were performed in cell-repellent treated 96-well plates (Greiner Bio-One 07-000-640). We selected the drug library from the SelleckChem Immunology and Inflammation-related library (L4100), or the FDA-approved drug library (L2000), which were supplied at a stock concentration of 10mM. The layout for drugs and/or drug combinations for each 96-well plate was pre-defined before starting the experiment. The final working concentrations for drugs varied between 0.01 nM to 10 uM. To stimulate PBMCs into an activated state, some wells included the use of ImmunoCult Human CD3/CD28 T Cell Activator (Stemcell Technologies 10971), which was used at the commercially recommended final concentration of 5 uL/200uL. After completing drug dilutions, we dispensed drugs into a 96-well plate layout, loaded 200,000 PBMCs into each well, and cultured for 24 hours in a tissue culture incubator at 37 °C and 5% CO2. All working drug dilutions and cell suspensions were prepared in RPMI-1640 + 1% Pen/strep + 10% FBS. The final concentration of DMSO within these experiments was always below 0.1%, which has been shown to produce minimal effects on immune cell activation [23].

## 1.12 High-throughput single-cell profiling using sample multiplexing techniques

After 24 hours of drug exposure, we harvested cells for single-cell profiling. Cells were enzymatically dissociated into a single-cell suspension and multiplexed using MULTI-seq lipid-modified oligos [5]. One experiment (MULT-19) was performed using 10X Chromium Cellplex Kit (1000261), and manufacturer protocols were followed. Although plates were treated to be cell repellent, activation of certain cell types within the PBMC mixture can produce strong adhesive tendencies to the plate and each other. Thus, we enzymatically digested both the cell suspension and the plate surface with 100 uL of TrypLE (Thermo Fisher 12604021) to ensure the complete recovery of all cells. Briefly, the digestion was performed as follows: Using a multichannel pipette, all cells from each plate were transferred to a conical V-bottom 96-well plate (Thermo Fisher 249935), and then 100 uL of TrypLE was added to each well of the original plate and kept at room temperature. The conical V-bottom well plate, which contains the cells originally in suspension, was centrifuged at 400 rcf for 4 minutes. The supernatant was removed, and 100 uL of TrypLE was added to each well of pelleted cells and triturated gently. Both the original plate and the v-bottom plate were then incubated at 37 °C for 2 minutes simultaneously. After checking that any adherent cells were removed from the original plate, the cells in each plate were triturated and pooled back together. Cells were then labeled with MULTI-seq lipid-modified oligos according to the original protocol [5] and profiled using the 10x Chromium 3' chemistry (v3.1, PN-1000121) or using the HT version (HT Kit v3.1, 1000370). Cells were super-loaded onto each lane of a Chromium chip at 25,000 - 40,000 cells per lane, or onto a Chromium HT chip at 60,000 cells/lane. Sample tags were purified and amplified according to the MULTI-seq procedure or according to manufacturer instructions for Cellplex. Libraries were sequenced using Illumina Novaseq S4 chips at the UCSF CAT Sequencing facility to a target depth of 40,000 transcriptome reads/cell barcode and 5000 sample tag reads/cell barcode. Cells that were multiply labeled by sample tags were computationally removed from the analysis.

### 1.13 Time-course profiling of T-cell mediated immune activation

A reverse time course of CD3/CD28 T Cell Activator was performed using a Tecan Evo robot (Caltech Protein Expression Center). Briefly, cryopreserved PBMCs from healthy donors were thawed and rested for 16 hours as described above. Cells were then pelleted and dissociated with gentle trituration and seeded into 96-well plates at 200,000 cells/well in 150 uL of RPMI-1640 + 10% FBS + 1% P/S. The cells were then brought to a Tecan Freedom Evo2 liquid handling robot, which was programmed to dispense CD3/CD28 activator into each well at 30-minute intervals for 30 hours. The CD3/CD28 activator was diluted 1:10 in RPMI-1640 + 1% Pen/strep + 10% FBS, and 50 uL of the dilution was added at each time point. One well which did not receive any activator served as timepoint 0. Between dispenses, the plate was held in an onboard incubator at 37 °C and 5% CO2. After the conclusion of the time course, cells were immediately harvested for single-cell profiling using multiplexing tags (MULTI-seq) as described above.

### 1.14 Alignment and de-multiplexing

We used the Cell Ranger software (10x Genomics) to align single-cell RNA sequencing reads to the transcriptome. For experiment batches MULT-6, 7, 8, 9, 10, 12, 13, 14, 15, 16, TC, we used Cell Ranger 3.1.0. For experiment batches MULT-17, and 18, we used Cell Ranger 5.0.1. For experiment batches MULT-19 and MULT-20, we used Cell Ranger 6.1.2. The 10x Genomics hg19-1.2.0 genome build was used as a reference transcriptome.

Except for MULT-19, where we used the 10X Cellplex pipeline to demultiplex the conditions of each cell, we used a custom Python pipeline to assign cells to each condition based on the sequenced tags following the pipeline in MULTI-seq [5].

## 2 Quantification and statistical analysis

### 2.1 Filtering, normalization, and clustering of the Perturb-seq dataset

The Perturb-seq dataset is a genome-wide screening study where every single gene is knocked down individually [24]. In Perturb-seq, a library of guide RNAs is introduced by lentiviral transduction into engineered cells that express the dCAS9 CRISPRi effector, after which single RNA sequencing (scRNA-seq) is conducted to simultaneously read out the transcription profile and guide RNA for each cell. Because guide RNA transfection is a stochastic process, the number of cells receiving each perturbation varies vastly. Here, the dataset contains perturbations with a sample size ranging from 2 to 2,000 cells. To reduce noise introduced by the small sample size, we excluded perturbations with less than 20 cells (metadata entry "number of cells (filtered)"), which represented 92 perturbations, 0.86% of all perturbations.

More than half of the perturbations do not yield significant single gene-level expression changes (apart from the perturbed gene), possibly due to genome redundancies that ensure the robustness of cellular activities, and that some genes may not function in the profiled experimental condition. To avoid the model overfitting the measurement noise on these perturbations with no significant effect, we excluded perturbations with less than 10 differentially expressed genes (metadata entry "Number of DEGs (anderson-darling)"), which represented 7,523 perturbations, 70.5% of all perturbations. Similarly, we only selected a subset of the 585 control samples with non-targeting guide RNAs to avoid overfitting on measurement noise. We selected control samples that were labeled as core control (metadata entry "core_control") and had no less than 200 collected cells (metadata entry "number of cells (filtered)"). In total, we selected 3,136 perturbation conditions and 105 control conditions.

Normalization of single-cell gene expression data is required because (1) the total number of transcripts captured for each cell can vary from 1,000 to 100,000 due to technical variability in reagents and library preparation steps, and (2) transcript expression levels across genes can span 5 orders of magnitude. We normalized the raw counts for each cell by the total counts detected in each cell, times a scaling factor that is roughly the median of the total counts of transcripts in a cell. We used the scaling factor $10^4$ as the target total count, that is, computing $g_i' = 10^4 * g_i / \sum_i g_i$ for every cell, where $g_i$ is the count of gene $i$. We also used a log transformation to mitigate the great variety of expression levels across genes $g_i'' = \log(g_i' + 1)$. Except for gene filtering, all the following analysis was performed on the log-transformed data.

Due to measurement noise and gene drop-out, genes that have constant expression across cells do not contribute to biological impact, but only increase noise in the data. Therefore, single-cell data analyses are performed on a subset of genes that have different expression levels across cells, i.e., highly variable genes. Here, we developed a highly variable gene filtering model adapted from the coefficient-of-variation (CV) filtering using the zero-inflated Poisson distribution as the statistics of gene counts. The model was motivated by our observation that genes with high expression generally have higher CV than the prediction of the Poisson model, and can be explained by modeling drop-out during sequencing. Drop-out means every gene count has a random probability of being mistakenly sampled as 0, which is also called zero inflation in statistics. For a

zero-inflated Poisson distribution,

$$P(X = x) = \begin{cases} \pi + (1-\pi)e^{-\lambda}, & \text{if } x = 0, \\ (1-\pi)e^{-\lambda}\lambda^x/x!, & \text{if } x > 0, \end{cases} \tag{47}$$

where $\lambda$ is the parameter of Poisson distribution and $\pi$ is the probability of drop-out or zero inflation. The mean and CV of the distribution is

$$\mu = (1-\pi)\lambda \tag{48}$$

$$\frac{\sigma}{\mu} = \frac{\sqrt{1 + \pi\mu/(1-\pi)}}{\sqrt{\mu}} \tag{49}$$

The zero-inflated Poisson distribution suggests another type of log CV ($y$) - log mean ($x$) relationship where the expected log CV is higher than the prediction of a Poisson model for high-expression genes.

$$y = -\frac{1}{2}\left[x - \log(1 + \frac{\pi}{1-\pi}\exp x)\right] \tag{50}$$

Therefore, we followed the form of the equation and obtained the empirical log CV - log mean relationship by curve fitting (scipy.optimize.curve_fit) of the following function

$$\hat{y} = -k\left(x - \log(1 + \pi\exp x)\right) + b \tag{51}$$

where we denoted $\pi/(1-\pi)$ into a single parameter $\pi$ for numerical stability. The genes are ranked by $\log \mathrm{CV} - \hat{y}$, the difference between log CV and curve fitting. A higher value indicates that the gene is more variable.

We selected highly-variable genes as the union of the following three sets: (1) we performed the Wilcoxon test of log-transformed data for each guide RNA and control (sc.tl.rank_genes_groups) and selected the top 10 differentially expressed genes for each guide that passed $p < 0.05$ under Bonferroni multi-test correction (we selected less than 10 if no enough genes pass the test), (2) the top 1,000 highly-variable genes selected by our zero-inflated Poisson CV filtering, and (3) for comparison with the original Perturb-seq study, we included the genes that appeared in the gene programs in the original Perturb-seq study. In total, we selected 3,273 genes for the analysis using the D-SPIN framework.

To visualize and cluster single cells in the Perturb-seq dataset, we corrected the batch effect by constructing latent representations with scVI on raw gene counts (dispersion='gene-batch', n_latent=10, n_layers=2, max_epoch=20, batch_size=2048) [25]. We constructed a 15-nearest-neighbor graph (scanpy.pp.neighbors) on the latent representation and computed Leiden clustering for cell clusters (scanpy.tl.leiden). As K562 is a purified cell line, it contains primarily a single cell type. Thus, the UMAP embedding of the cell population is a circular structure, representing the progression of the cell cycle, which is less relevant to the study of perturbation responses. Therefore, we only show the UMAP embedding on control samples along with D-SPIN-generated samples using cell-cycle-associated gene programs (Sec. 2.13).

## 2.2 Filtering, normalization, and clustering of the drug profiling dataset

The single-cell gene expression normalization in the drug profiling dataset is generally identical to the Perturb-seq dataset. We normalized the raw counts by each cell with the same scaling factor $10^4$, and gene expressions were log-transformed for the following analyses except for the highly variable gene selection.

The highly variable gene selection was based on the zero-inflated Poisson expression model we developed (Sec. 2.1). The drug profiling dataset contains (1) a large dataset of single drug and drug combination profiling (Drug Profiling) and (2) a dataset of follow-up drug dosage combinations (Dosage Combination). The gene filtering was performed on the Drug Profiling dataset, and Dosage Combination follows the same filtering. To avoid noise in low-expression genes, we excluded genes that only expressed in less than 0.1% of the cells in the Drug Profiling dataset. (In the Perturb-seq dataset, the low-expression genes were already filtered out by the authors in the published data). We selected 3,287 highly-variable genes in the remaining genes, which is $\log 10(1.1)$ above the curve fitting Eq. 51 for the zero-inflated Poisson distribution model.

Collected single-cell profiling data have batch effects across experiments, likely due to variations in cell culture conditions and technical differences across sequencing machine runs. To visualize, cluster, and cell-typing single-cells in the Drug Profiling dataset, we corrected batch effects by constructing latent representations on raw gene counts of 5,000 selected genes (scanpy.pp.highly_variable_genes) with scVI (dispersion='gene-batch', n_latent=10, n_layers=2, max_epoch=200, batch_size=5120) [25]. We constructed a 15-nearest-neighbor graph (scanpy.pp.neighbors) on the latent representation, computed Leiden clustering for cell clusters (scanpy.tl.leiden), and conducted UMAP embedding for visualization (scanpy.tl.umap). The cell subtype identities of each Leiden cluster were annotated manually by comparing the most significant up-regulated genes of the cluster with the

remaining cells. We removed 3 clusters that mostly came from the experimental batch "MULT-6" and were putatively cell debris, as they expressed the maternal effect gene OOEP and disrupted transcriptional states were present. We also excluded red blood cells (RBC) and plasma B-cells for D-SPIN analysis, as they are small and isolated cell populations with distinct transcriptional profiles.

The Dosage Combination dataset followed the same normalization and log-transformation pre-prossessing. We visualized the data by batch effect correction with scVI (dispersion='gene-batch', n_latent=10, n_layers=2, max_epoch=100, batch_size=5120) followed by the same neighborhood graph, Leiden clustering, and UMAP embedding procedures.

## 2.3 oNMF decomposition of single-cell dataset

In the Perturb-seq study, we used a subset of $10^5$ single cells, with the Square-root balancing scheme in Sec. 1.8 for the Leiden cluster of cell states. In the drug profiling study, we used a subset of $3.3 \times 10^5$ single cells using the equal-balancing scheme, sampling 1,000 single cells from each Leiden cluster of cell states. In drug profiling data, the sampling scheme was to better capture gene programs in drug-specific cell states that composed a small proportion of all the cells.

In the Perturb-seq study, we ran oNMF 100 times with $K = 30$ with different random seeds and computed a consensus gene program set by K-means. In the drug profiling study, as specific drug-induced cell states were a small fraction of the whole dataset, we increased the resolution of oNMF to $K = 50$. During preliminary runs of oNMF, we identified drug-specific response programs such as Pathogen response and Stress response. We then manually curated random seeding to select the oNMF decomposition that included these target gene programs. We then combined the 50 programs into 30 by merging programs with similar biological functions and distribution of cell subtype expression.

## 2.4 Inferring regulatory networks and perturbation responses with D-SPIN

**Sample selection** In the Perturb-seq study, a large proportion of perturbations induce insignificant effects on the gene program level. To avoid overfitting on measurement noise, we selected a subset of 128 samples that were most different from each other (Sec. 1.1). The subset of samples was selected by clustering all samples into 128 groups by their cross-correlation, and took one sample from each group. We then learned the response vectors of all other samples independently conditioned on the inferred network. In the HSC network, large synthetic network, and drug profiling study, we infer the regulatory network with all experimental conditions in the dataset.

**Gradient evaluation** We used exact maximum likelihood inference for the HSC dataset (11 nodes), MCMC maximum likelihood inference for Perturb-seq and drug profiling dataset (30 nodes), and pseudo-likelihood method for the large synthetic network dataset (125 $\sim$ 1,000 nodes). In MCMC of the 30-node network, we used $5 \times 10^7$ samples in each iteration for each condition with $10^5$ samples warm-up for network inference and $10^7$ samples for perturbation response vector inference.

**Regularization** In the HSC network dataset, no regularization was used to faithfully benchmark the network inference accuracy. In the Perturb-seq and drug profiling dataset, we used $\ell_1$ regularization on the network with $\lambda_1 = 0.01$, promoting the sparsity and interpretability of the network. We used $\ell_2$ regularization on the response vectors with $\lambda_2 = 0.005$ to avoid an excessively large magnitude of response with strong activation and inhibitions. No priors on the perturbation response were used.

In the large synthetic network, we used $\ell_1$ regularization on the network with $\lambda_1 = 0.01$. As the perturbation is known, we applied a prior on the perturbation response as described in Sec. 1.4. Note that the perturbation actions were changing the coefficient of stochastic differential equation (SDE) models of biochemical reactions (Sec. 2.8), which is a totally different type of model from the probabilistic graphical models in D-SPIN. Nonetheless, the information about the perturbations could still assist the network inference.

Specifically, to align the strength of perturbation impact across the SDE and D-SPIN models, we compared the gene expression distribution change under single-gene knockdown and activation in a three-node toy model. We found that scaling the numerical values of perturbation action in SDE by a factor of 1.5 best correlates with their effect in D-SPIN models. Therefore, we set the prior perturbation response $\boldsymbol{h}_r^{(n)}$ to be 1.5 times the magnitude of SDE perturbations, and used a regularization coefficient $\lambda = 1$.

**Numerical optimization** In optimization, we used the Adam optimizer with a learning rate of 0.2 for exact maximum likelihood, 0.02 for MCMC maximum likelihood, and 0.05 for pseudolikelihood. The decay rate was the default of Adam optimizer as $\beta 1 = 0.9,\ \beta 2 = 0.999,\ \epsilon = 10^{-6}$. Adam optimizer can significantly speed up the convergence rate of optimization, but suffers from instability during the final stage of optimization. We therefore deploy a backtracking scheme where we roll back 20 iterations and half the step size when

the current gradient is larger than twice the gradient 20 iterations before. We used 200 $\sim$ 1000 learning iterations, depending on the convergence rate, and selected the network with the smallest gradient in the learning trajectories.

## 2.5 Parallelization of D-SPIN inference

An advantage of D-SPIN is that we can parcel out computing jobs to different CPUs. For instance, the gradient estimation of each summation term in Eq. 7 and each condition in Eq. 8 are independent of each other. Leveraging the property, D-SPIN performs parallelized model inference by parceling computation for each condition to different CPUs. We use the MATLAB "parfor" function on the Caltech Resnick cluster, with each task taking 2 cores and 32G memory. The inference of response vector $\boldsymbol{h}$ given on a fixed network $\boldsymbol{J}$ is also independent for each condition and is parallelized with the Slurm Workload Manager system on the Caltech Resnick cluster with 2 cores and 32G memory.

## 2.6 Simulations of the HSC network

The BEELINE framework provides a systematic tool to simulate single-gene expression data from Boolean regulatory networks and to benchmark the accuracy of network inference methods [26]. Our simulation and evaluation mostly followed the formulation and code of BEELINE, but we complemented the framework with the simulation of gene knockdown and activation perturbations. BEELINE simulates the following stochastic differential equation (SDE) models of gene expression.

$$\frac{d[x_i]}{dt} = m_i f(R_i) - l_x[x_i] + s\sqrt{[x_i]}\Delta W_t \tag{52}$$

$$\frac{d[p_i]}{dt} = r[x_i] - l_p[p_i] + s\sqrt{[p_i]}\Delta W_t \tag{53}$$

$$\Delta W_t = \mathcal{N}(0, h) \tag{54}$$

where $[x_i]$ and $[p_i]$ are mRNA and protein concentration of the gene $i$, $m_i$ and $r$ is the mRNA and protein production rate, $f(R_i)$ is the regulatory term determined by the network, $l_x$ and $l_p$ are mRNA and protein degradation rate, $s$ is thermal noise strength, $W_t$ is the Wiener process representing thermal noises, and $h$ is simulation time step. The default parameter values are $m_i = 20, l_x = 10, r = 10, l_p = 1, s = 10$. At $t = 0$ all mRNA concentrations $[x_i]$ are initialized as 1, and all protein concentrations $[p_i]$ are initialized as 0. Specifically, the mRNA default initial concentration 1 in the current BEELINE framework (v1.0) is different from 0.01 used in the version (v0.1) presented by the BEELINE paper [26], possibly for improved numerical stability.

To model the impact of gene knockdown and activation, we included two extra terms into the dynamics of mRNA: $e_i$, representing a constant external mRNA production not regulated by the network, and $\beta_i$, representing the proportion that the mRNA production rate is suppressed in addition to the network regulation. After the modification the dynamics of the mRNA is

$$\frac{d[x_i]}{dt} = e_i + \beta_i m_i f(R_i) - l_x[x_i] + s\sqrt{[x_i]}\Delta W_t \tag{55}$$

Without perturbation, we have $e_i = 0, \beta_i = 1$.

In single gene and double gene knockdown simulation, we used $\beta_i = 0.2$ for the reduced mRNA production rate. In single gene activation simulation, we set $e_i = 15$, corresponding to 0.75 of the expression of fully activated genes. For double perturbations, we found that higher external activation was required to drive the network, and we used $e_i = 22.5$ for gene activation in double perturbations. We simulated 2,000 cells for control conditions and 500 cells for each perturbation condition.

We simulated the system under 10 different sets of randomly sampled parameters. Each time the parameters $m_i, l_x, l_p, r$ were assigned by the formula $\alpha D$, where $D$ represents the default value of the parameter and $\alpha$ is a random factor sampled from a truncated Gaussian distribution (mean = 1, std = 0.1, truncation to $[0.9, 1.1]$). The same type of parameters of all genes were set to the same value for numerical stability. We clustered the cell states and computed the UMAP embedding of the discretized simulated data with D-SPIN-generated samples for visualization, and annotated each cluster based on the research that proposed the HSC network model [12].

To compare the network reconstruction accuracy with other methods, including PIDC, GRNBoost2, and GENIE3, we ran the Docker images of the three methods provided by the BEELINE framework. As the three methods do not explicitly model the impact of perturbations, we ran these methods independently on control and each perturbation condition and computed the average score of each network edge as the overall inferred network.

## 2.7 Clamping analysis of node marginal distribution and edge sensitivity

As a generative model of transcriptional state distribution, D-SPIN allows us to develop a procedure to compute perturbation impact on the marginal distribution of a node in the network, as well as the contribution of each interaction edge to the marginal distribution. The procedure allows us to identify critical interactions in the network that determine a specific perturbation response so that we can make quantitative statements about how cell state distribution and perturbation response are determined by the regulatory network.

To quantify the marginal distribution change of each node $i$, we define an effective marginal activity $\hat{h}_i(\boldsymbol{J}, \boldsymbol{h})$ by the ratio between the node being on ($s_i = 1$) and the node being off ($s_i = -1$) in the marginal distribution $P(s_i|\boldsymbol{J}, \boldsymbol{h})$

$$\hat{h}_i(\boldsymbol{J}, \boldsymbol{h}) = \log \frac{P(s_i = 1|\boldsymbol{J}, \boldsymbol{h})}{P(s_i = -1|\boldsymbol{J}, \boldsymbol{h})}\,, \tag{56}$$

where positive $\hat{h}_i$ indicates the node being activated and negative $\hat{h}_i$ indicates the node being repressed. The definition is rationalized by parametrizing the complete three-state marginal distribution of node $i$ by an effective 1-node D-SPIN model

$$P(s_i) = \frac{1}{Z} \exp(\hat{J} s_i^2 + \hat{h}_i s_i)\,, \tag{57}$$

whose analytical solution is $\hat{h}_i = \frac{1}{2} \log P(s_i = 1)/P(s_i = -1)$, and we drop the constant coefficient for simplicity. As $\hat{h}_i$ is defined as a linear term inside the energy function, the difference of $\hat{h}_i$ between conditions represents the relative activation or inhibition of a node.

To quantify the contribution of each edge to the marginal distribution change, we compare the marginal distribution with and without the edge of interest. The edge sensitivity of $J_{kl}$ to node $i$ is defined as the marginal activity $\hat{h}_i$ change after removing the edge $J_{kl}$ from the network. Specifically, we define a clamped network $\boldsymbol{J}_{(\text{set} J_{kl}=0)}$ where $J_{kl}$ is set to 0 while all other edges remain the same, and compute the marginal activity $\hat{h}_i(\boldsymbol{J}_{(\text{set} J_{kl}=0)}, \boldsymbol{h})$ of node $i$. The edge sensitivity $\epsilon(kl, i)$ is thus defined by

$$\begin{aligned}\epsilon(kl, i|\boldsymbol{J}, \boldsymbol{h}) &= \hat{h}_i(\boldsymbol{J}, \boldsymbol{h}) - \hat{h}_i(\boldsymbol{J}_{(\text{set} J_{kl}=0)}, \boldsymbol{h}) \\ &= \log \frac{P(s_i = 1|\boldsymbol{J}, \boldsymbol{h})}{P(s_i = -1|\boldsymbol{J}, \boldsymbol{h})} - \log \frac{P(s_i = 1|\boldsymbol{J}_{(\text{set} J_{kl}=0)}, \boldsymbol{h})}{P(s_i = -1|\boldsymbol{J}_{(\text{set} J_{kl}=0)}, \boldsymbol{h})}\end{aligned} \tag{58}$$

With the HSC network example, we demonstrate how this procedure identifies the impact of Pu1 knock-down on the node Gfi1, and which network edge contributes to the change of Gfi1. D-SPIN infers the network $\boldsymbol{J}$, response vector $\boldsymbol{h}_0$ without perturbation and $\boldsymbol{h}$ under Pu1 knock-down. Without perturbation, by marginalizing the full distribution $P(\boldsymbol{s}|\boldsymbol{J}, \boldsymbol{h}_0)$, we have

$$P(\text{Gfi1} = -1) = 0.884, P(\text{Gfi1} = 1) = 0.024, \text{ thus } \hat{h}_0 = -3.60\,.$$

Similarly, under Pu1 knock-down, marginalizing the complete distribution $P(\boldsymbol{s}|\boldsymbol{J}, \boldsymbol{h}_0)$ gives

$$P(\text{Gfi1} = -1) = 0.381, P(\text{Gfi1} = 1) = 0.216, \text{ thus } \hat{h} = -0.566\,.$$

Therefore Pu1 knock-down has an overall impact of 3.04 activation on Gfi1. For edge sensitivity, take the edge Gfi1-EgrNab, for example. After clamping the edge to 0, the complete distribution $P(\boldsymbol{s}|\boldsymbol{J}_{(\text{set Gfi1-EgrNab = 0})}, \boldsymbol{h})$ has an marginal distribution of

$$P(\text{Gfi1} = -1) = 0.904, P(\text{Gfi1} = 1) = 0.004, \text{ thus } \hat{h}_{(\text{set Gfi1-EgrNab = 0})} = -5.26\,.$$

Therefore the edge sensitivity $\epsilon(\text{Gfi1-EgrNab}, \text{Gfi1}) = \hat{h} - \hat{h}_{(set\text{Gfi1-EgrNab=0})} = 4.69$. The edge Gfi1-EgrNab has a high contribution in activating the node Gfi1. The contribution of other edges can be computed similarly, and the top contributing edges to Gfi1 under Pu1 knock-down are Gfi1-EgrNab 4.69, Cebpa-Gfi1 -2.85, Gfi1-Pu1 -2.60 and Gfi1-cJun 2.31.

## 2.8 Simulation of large synthetic networks

To assess the accuracy and scalability of D-SPIN, we constructed two distinct sets of large regulatory network models, each comprising networks with 125, 250, 500, and 1000 nodes. These models represent two major categories of networks:

1. Modular Networks: These networks are organized into modules, characterized by activating interactions inside each module and inhibitory interactions between modules. Consequently, gene expression of these networks also exhibits a modular pattern, where genes within the same module are more likely to be activated at the same time.

2. Random Networks: One of the most representative random networks are Erdős–Rényi models (ER models). These networks are completely random in nature, where each node has the same probability of interacting with any other node.

Specifically, in our construction of modular networks, the networks contain $Q$ modules that form a loop by connecting neighboring modules on a circle. The average module size is set to around 50 nodes, and the total number of modules should be an odd number to allow the pattern of alternatively activated modules on the loop. Therefore the module numbers $Q$ for 125, 250, 500, and 1000-node networks are, respectively, 3, 5, 11, and 21. The exact number of nodes in each module is determined by sampling Poisson random numbers, with the distribution expectation being the expected module size. Within each module, each node has a 0.5 probability of being assigned as a core node. Each core node has a 0.0375 probability of activating another node in the same module, and a 0.05 probability of inhibiting another core node in neighboring modules. Each non-core node has a 0.0125 probability of activating another node in the same module. Besides, following the structure of the classical toggle switch architecture, all inhibitory interactions are symmetric, and core nodes that have inhibitory interactions also have self-activating interactions.

The construction of a random network is straightforward. To ensure the same average degree of network nodes across different network sizes, each node has a $1.25/M$ probability to activate any other nodes and $1.25/M$ probability to inhibit any other nodes, where $M$ is the number of nodes in the network. We ensure the network is fully connected by regenerating the network with a different random seed when the network has a disconnected structure.

Using these synthetic networks, we simulated the dynamics of each model with our modified BEELINE framework. Given the complexity and scope limitations of designing combinatorial perturbations on large networks, we opted for random perturbations, where each node was subject to a random perturbation following a truncated Gaussian distribution. For each node $i$, the perturbation $h_i^*$ was taken from a Gaussian distribution with a mean of 0, a standard deviation of 0.4 and truncated to $[-1, 1]$. These perturbations dictated the biochemical parameters in the simulation by

$$e_i = \begin{cases} 20h_i^* & h_i^* > 0, \\ 0 & h_i^* \leq 0. \end{cases} \quad \beta_i = \begin{cases} 1 + h_i^* & h_i^* < 0, \\ 0 & h_i^* \geq 0. \end{cases} \tag{59}$$

For a perturbation $h_i^* = 1$, the strength of external activation is the same as the strongest internal expression in the network. For a perturbation $h_i^* = -1$, the gene $i$ is completely deactivated. Therefore this perturbation assignment covered a spectrum of perturbation actions for accessing the capability of D-SPIN in constructing regulatory networks from perturbations.

## 2.9 Benchmarking time consumption with cell number

To assess the scalability of each method relative to cell number, we performed time benchmarks for D-SPIN together with PIDC, GENIE3, and GRNBoost2, using the dataset of 10 different random repeats of 250-node Erdős–Rényi networks from Sec. 2.8. Specifically for D-SPIN, the gradient evaluations were performed by batches for efficient computation, with each batch containing 2,000 cells. For PIDC, GENIE3, and GRNBoost2, we used the Docker images of the three methods provided by the BEELINE framework. The analyses were performed on the Caltech Resnick cluster, with each task taking 2 cores and 48G memory. The longest running time of each task was 7 days. The time of unfinished tasks was interpolated using the running time of the same methods with smaller cell numbers using the function $y = Ax^B + C$ in the log-log space.

## 2.10 Evaluating single-gene network reconstruction with ChIP-seq data

To further assess the performance of D-SPIN on regulatory network reconstruction with biological measurement datasets, we applied different network inference methods to the genome-wide Perturb-seq dataset, and compared the results with biological validations obtained from ChIP-seq data. Chip-seq data measure the binding of transcription factors (TFs) to target gene promoters and have been used in the literature to assess the quality of inferred gene regulatory network models. We evaluate the correspondence between inferred network and ChIP-seq data by the number of inferred edges that can be associated with a measured TF-target gene binding event.

Specifically, we generated two datasets from the Perturb-seq data, similar to the BEELINE framework [26], labeled as TFs+500 and TFs+1000. We selected TFs that expressed in over 5% of cells and the top 500 or 1000 differentially expressed genes that were identified with the zero-inflated Poisson distribution model we developed (Sec. 2.1. We only included genes that had corresponding knockdown perturbation conditions that satisfied the following criteria.

1. More than 20 cells collected (metadata entry "number of cells (filtered)").
2. More than 2 differentially expressed genes (DEGs) identified (metadata entry "Number of DEGs (anderson-darling)").

3. A knockdown efficiency exceeding 5% (metadata entry "percent knockdown").

Additionally, we incorporated 46 control conditions with non-targeting guide RNA, with a minimum of 250 cells each. After filtering, the TFs+500 dataset comprised 103k cells, 421 genes, and 467 conditions, while the TFs+1000 dataset included 150k cells, 624 genes, and 670 conditions.

For existing network inference methods, considering the scalability challenges faced by PIDC, GENIE3 and GRNBoost2 with datasets exceeding 100k cells, as previously benchmarked in Sec. 2.9, we divided the cells into approximately 2k-cell chunks. Each method was then applied to these chunks independently, with the resulting networks averaged to produce an overall inference.

For D-SPIN, we partitioned the data based on each perturbation condition. We did not partition by condition for other methods, because many perturbation conditions only had a few tens of cells, which would be insufficient for their network inference. D-SPIN can handle perturbations with small cell numbers as information is shared through inferring a unified regulatory network. As we knew the target of each knockdown perturbation, we used the prior $h_i^* = -3$ for the knockdown target genes. We used pseudolikelihood inference because of the large network size. We applied $\ell_1$ regularization on the network with $\lambda_1 = 5e - 3$, and perturbation prior regularization with $\lambda = 0.5$. Compared with the inference of the large synthetic network (Sec. 2.4), we used a smaller perturbation prior regularization coefficient, as the effect of experimental single-gene knockdown is more noisy and uncertain.

The inferred networks were compared with processed ChIP-seq data of TF-target interactions from human ChIP-seq datasets. We used the interactions from DoRothEA [27], RegNetwork [28], and TRRUST [29], which were integrated together by the BEELINE framework. As the ChIP-seq data only contained TF-target regulations, we removed all inferred interactions between non-TF genes, which was consistent with the BEELINE framework. The correspondence with ChIP-seq data is quantified by the early precision rate (EPR), which measures the proportion of inferred edges that can be associated with a TF binding event detected by ChIP-seq among the top $K$ predictions, relative to the expectation of a random predictor. $K$ is set to the total number of TF-target interactions in the ChIP-seq database.

## 2.11 Discovering regulatory networks modules and clustering perturbation responses

To gain insight into the structure and modular organization of the regulatory networks inferred by D-SPIN, we applied the Leiden algorithm for graph community detection to both Perturb-seq and drug profiling datasets. The Leiden algorithm identifies densely connected subnetworks, or communities, by optimizing a modularity score that reflects the density of positive interactions within communities compared to a random network. For networks with both positive and negative interactions, the following score function is employed [30].

$$Q = Q(J_+) - Q(J_-) \tag{60}$$

$$Q := \frac{1}{2m} \sum_{i,j} \left[ \left( J_{ij} - \gamma \frac{k_i k_j}{2m} \right) \delta(c_i, c_j) \right] \tag{61}$$

where the regulatory network is split into $J_+$ for positive and $J_-$ for negative interactions. In the score function $Q$, $m$ is the sum of all edge weights, $k_i$ denotes the sum of weights for edges connected to node $i$, $c_i$ is the community to which $i$ belongs, and $\gamma$ is the resolution parameter. We set $\gamma = 2$ to achieve a finer resolution of module discovery, while the presented result is robust with a range of $\gamma$ from 1.05 to 2.5.. The Leiden algorithm partitions the network into distinct communities, whose biological functions were inferred from the gene programs they contained.

In addition to analyzing the network architecture analysis, we also clustered the perturbation response vectors to uncover interaction patterns between perturbations and the network. In general, we computed relative response $\tilde{\boldsymbol{h}}$ for each $\boldsymbol{h}$ by subtracting the average $\boldsymbol{h}$ of control samples. Effective perturbations were then selected by comparing with the distribution of relative responses in control samples.

In the Perturb-seq dataset, we selected 1,755 effective perturbations with $p < 0.05$ under Bonferroni multi-test correction using non-targeting guide RNA as null distribution. We clustered perturbations into groups, because the perturbation impacts are noisy due to the low cell number in each condition and varying knockdown efficiencies. We used the Leiden algorithm for better clustering of the noisy perturbation responses. We built an 8-nearest-neighborhood graph and ran Leiden algorithms with resolution $\gamma = 3$ to partition the perturbation response vectors into 40 groups. The relative responses of guide groups were defined as the average relative responses in the group.

In the drug profiling dataset, due to some batches only having a single control sample, a unified statistical test for the significance of perturbation response vectors is not viable. We scored each drug by the negative log probability density function at $\tilde{\boldsymbol{h}}$ of a multivariate Gaussian distribution with zero mean and diagonal covariance. The covariance was determined by the variance of control samples of the same experimental batch if there were more than 1 control sample, otherwise by the variance of all control samples in the dataset.

We selected 158 candidate drugs with a score above 20, and performed K-means clustering with 20 clusters. We found that some drug clusters were putative experimental batch effects, as these contained biochemically different drugs from the same experimental batch. We also removed the drug cluster where a complete loss of macrophage population was observed, as the loss of macrophage population was not reproducible in other experiments, and likely resulted from cell adhesion to the culture plate. The remaining drug clusters were then manually reviewed and consolidated based on their biological impacts, resulting in 70 drugs in 7 major categories.

## 2.12 Automated identification of regulatory strategies and network coarse-graining

The response vectors can further reveal the information processing of the regulatory network by elucidating how perturbation responses are organized across network modules. The response vectors not only identify perturbation impacts on individual programs, but also provide a global view of how cells respond to the stress induced by the perturbation. To further consolidate perturbation groups, typical measures such as Euclidean distance are not sufficient, as each gene program is not an independent dimension but interacts closely with other gene programs through the regulatory network. Therefore, we used a graph clustering strategy to integrate information of program interactions into the clustering.

Specifically, we built an extended network by also including perturbations as nodes of the network. The extended network encompassed both interactions between gene programs and interactions between perturbation and programs. Formally, the adjacency matrix of the extended network is

$$\boldsymbol{J}_{\text{ext}} = \begin{bmatrix} \boldsymbol{J} & [\tilde{\boldsymbol{h}}_1, \tilde{\boldsymbol{h}}_2, \dots] \\ [\tilde{\boldsymbol{h}}_1, \tilde{\boldsymbol{h}}_2, \dots]^\intercal & \boldsymbol{0} \end{bmatrix}, \tag{62}$$

where $\tilde{\boldsymbol{h}}$ is the average relative response for each perturbation group defined in Sec. 2.11. We then ran the Leiden algorithm on the extended network to detect perturbations that had similar impacts at the network level. In Leiden algorithms, modules of gene programs were fixed, and we only partitioned the perturbations. Perturbations were partitioned into the same community if they influenced gene programs that play analogous roles within the regulatory network. We set $\gamma = 1$ in the Leiden algorithm, while the algorithm always generates 4 perturbation clusters for $\gamma$ from 0.5 to 2.5, and the presented cluster assignment result is robust with a range of $\gamma$ from 0.9 to 1.45..

Through graph clustering, perturbation groups were assigned to modules in the regulatory network. Based on the score function Eq. 61 of the Leiden algorithm, this suggested that the perturbations predominantly activated gene programs within the module and were more likely to inhibit the gene programs outside the module. Conceptually, this correspondence between perturbations and modules established a connection of homeostasis regulation logic, that the cell upregulated the gene programs in the module in response to the loss of function caused by the perturbations. Therefore, we named the global strategy of perturbation responses according to the gene module that the perturbations were grouped with. For example, perturbations that were assigned to the Translation module were named "Translation up-regulation strategy".

We further visualized the global organization of biological functions and perturbation response strategies with a minimal network derived from the D-SPIN model by coarse-graining. In this simplified representation, both gene program modules and response strategies were condensed into single network nodes, as programs in the same module mostly have positive interactions inside and collectively contribute to a specific biological function. In the minimal network, the interactions between modules were determined by the most significant positive or negative interactions across their constituent gene programs. Similarly, the sign of the interaction between a gene module and a perturbation strategy was based on whether the activating or inhibiting interactions were stronger on average across the constituent gene programs and perturbation groups, and the strength is the average strength. As a visualization tool, the minimal network spotlighted the major response strategies employed by the network. For example, the impairment of translation machinery induced downregulation of the Translation module and upregulation of the Degradation module.

## 2.13 Embedding and comparison of D-SPIN and data distributions

To access the alignment between samples generated from D-SPIN and experimental data (or synthetic data as in Sec. 2.6 and Sec. 2.8), we employed several measures, including Uniform Manifold Approximation and Projection (UMAP), cell state distributions, distribution moments, optimal transport and exact probability distribution.

For small networks, the number of possible states $\boldsymbol{s}$ was relatively small and the distribution defined by D-SPIN could be explicitly computed, such as the HSC network and the cell-cycle subnetwork of the Perturb-seq data. We directly took samples from the D-SPIN distributions, computed joint embedding of D-SPIN samples and experimental data with UMAP, and computed Leiden clustering on the nearest-neighborhood

graph obtained during the computation of UMAP. As states in the model are discretized states $[-1, 0, 1]$, to avoid the neighborhood graph being disconnected due to frequent states forming cliques between themselves, we added a Gaussian noise on sample states with zero mean and 0.1 standard deviation.

For larger networks, the total $3^M$ possible states were impossible to enumerate, and the network encompassed a more complex landscape of cell states, such as the network of the drug profiling dataset. For a comprehensive evaluation of state distribution by D-SPIN and experimental data, we sampled $10^6$ samples by MCMC (Sec. 1.3) for each experimental condition as an empirical distribution of the D-SPIN model, and performed the following analysis.

1. UMAP embeddings of samples from the D-SPIN model. In the drug profiling dataset, there were more than 1,000 conditions and the total number of D-SPIN-generated states was more than $10^9$ ($10^6$ for each condition) and intractable for joint UMAP embedding. Therefore, we projected the states generated by D-SPIN to the batch-corrected UMAP of the experimental data with over 1 million cells (Sec. 2.2). Specifically, for each D-SPIN sample, we mapped it to the nearest neighbor of the discretized states in the experimental data. In the case of multiple nearest neighbors, the probability was evenly distributed between these cells. In this way, the $10^6$ samples from D-SPIN for each condition defined a distribution over the entire observed dataset, and could be visualized on the UMAP. We sampled $10^5$ cells from the distribution to render the UMAP for D-SPIN samples in Figure 5E.
2. Similarity of distribution on cell subtypes. The projected distribution over the whole dataset also defined a distribution over the cell subtypes as the clusters in the UMAP embedding. We computed the cosine similarity between cell subtype distributions of experimental data and the D-SPIN model for each condition. For reference comparison, we also defined two null distributions of cell state: the first null distribution is the uniform distribution over all possible states in the discrete space of $[-1, 0, 1]^M$, the second null distribution is the uniform distribution over the drug profiling dataset by pooling all conditions together to reflect the relative cell type abundance. The cosine similarity is presented in Figure 5F.
3. Gene program expression mean and cross-correlation. The first- and second-order moments of distributions are their fundamental characteristics. We computed the Euclidean (Frobenius) norm of the difference between the mean (cross-correlation) of the experimental data and D-SPIN model and presented in Figure S5F(i)(ii)
4. Optimal transport (OT) distance between experimental data and D-SPIN model. OT offers a robust method to quantify the distance between two probability distributions, taking into account the underlying geometry of the data space. For computational tractability, we computed the entropy-regularized Sinkhorn OT distance with python package POT (pot.sinkhorn(reg=0.5)) between experimental data and D-SPIN model distributions and presented in Figure S5F(iii). Specifically, as the samples of the D-SPIN model were obtained through MCMC, they were highly noisy at the single-state level. States that appeared only once or twice were likely by chance and led to skewed representations of the true distributions. This noise could be mitigated with neighborhood projection or distribution average in previous metrics, but had large impacts on state-level distribution comparisons such as OT. Therefore we filtered out states in the D-SPIN model that appeared fewer than 5 times, which were roughly half of the $10^6$ samples.
5. Direct state probability comparison. We further directly compared the probability of each single state, presented as scatter plots as in Figure S5G. The probability of the D-SPIN model also filtered out states that appeared fewer than 5 times, as in optimal transport distance computation.

## 2.14 Coherence of molecular complexes in CORUM

To obtain a comprehensive list of molecular complexes whose subunit knockdown effects were captured by the Perturb-seq dataset, we filtered the list of protein complexes in the CORUM database [4] by two criteria: (1) each complex should have no fewer than 10 subunits, and (2) at least 2/3 of subunits are contained in our filtered list of guide RNA, as in Sec. 2.1. Due to the presence of closely related complexes with many shared subunits in the database, we constructed a similarity graph using the overlap of subunits as edges. Through clustering, we distilled these into 46 distinct categories, and selected the most representative complex from each class that had the greatest number of subunits.

To assess the consistency of perturbation effects on subunits of each complex, we computed a coherence score as the average cosine similarity across all pairs of subunit perturbation response vectors.

$$\text{coherence} = \frac{2}{N(N-1)} \sum_{i<j} \frac{\tilde{\boldsymbol{h}}_i^\intercal \cdot \tilde{\boldsymbol{h}}_j}{\|\tilde{\boldsymbol{h}}_i\| \|\tilde{\boldsymbol{h}}_j\|}, \tag{63}$$

where $N$ denotes the number of subunits in the complex, and $\tilde{\boldsymbol{h}}_i$ represents the relative response vector for knockdown of subunit $i$.

The clustering of perturbation response vectors further revealed functional partitions of subunits within molecular complexes. Selecting an optimal number of clusters does not have a universal rule and is inherently subjective based on the context of the analysis. Among various metrics available to guide this choice, we employed the Silhouette score, which evaluates clustering quality based on the average distances within clusters and between neighboring clusters. A higher Silhouette score indicates a better quality of clustering.

For each complex, we performed K-means clustering, with the number of clusters $K$ ranging from 2 to 15, and calculated the Silhouette score for each. We considered a complex to exhibit distinct functional subunit groups if the peak Silhouette score was reached for a $K$ larger than 3. Examples of molecular complexes that exhibited modular organization included Proteasome, Integrator, Large Drosha complex, ALL-1 complex, and Spliceosome E complex, as presented in Figure 3G and Figure S3I.

## 2.15 Identification of drug target and potency from database

The drug target information was obtained by querying the probes & drugs database [31]. Repetitive entries in the database were consolidated with manual annotations. The primary target of drugs is based on Drug Bank [32] as well as manual lookups.

## 2.16 Computing strength of immune inhibitors

The strength of immune inhibitors in the D-SPIN model is quantified by projecting each inhibitor's relative response vector $\tilde{\boldsymbol{h}}$ onto a principal direction, denoted by $\boldsymbol{e}$ with the expression Strength $= \tilde{\boldsymbol{h}} \cdot \boldsymbol{e}$. The principal direction, $\boldsymbol{e}$, is derived from the leading left singular vector of the matrix composed of stacked relative response vectors $[\tilde{\boldsymbol{h}}_1, \tilde{\boldsymbol{h}}_2, \dots]$ from all immune inhibitors, including strong inhibitors, weak inhibitors I, and weak inhibitors II. The leading singular vector, determined through singular value decomposition (SVD), captures the primary axis of variance of the inhibitor response vectors. Essentially, it represents the consensus direction of gene program combinations that all immune inhibitors act on.

## 2.17 Categories of drug interactions on gene program level

We define 5 major types of drug interactions on the gene-program level: additive, subadditive, dominant, synergistic, and antagonistic. Denoting the response of two single drugs and drug combinations on a specific gene program as $h_1, h_2, h_c$, we classify different drug interactions based on the sign and magnitude of the single drug and combination responses. We define a tolerance $\delta = 0.5$, and the sign of a response $h$ is defined as the sign after soft thresholding

$$\operatorname{sgn}(h) = \begin{cases} -1 & \text{if } h < -\delta, \\ 0 & \text{if } -\delta \leq h \leq \delta, \\ 1 & \text{if } h > \delta. \end{cases} \tag{64}$$

Additive interactions are defined as

$$|h_c - h_1 - h_2| < \delta \tag{65}$$

Otherwise, if the two drug responses have the same sign $\operatorname{sgn}(h_1)\operatorname{sgn}(h_2) > 0$, we assume $h_1 > 0, h_2 > 0$ for notation simplicity and the situation of $h1 < 0, h_2 < 0$ naturally follows. The interactions are defined by

$$\text{interaction} = \begin{cases} \text{synergistic} & \text{if } h_c \geq \max(h_1, h_2) + \delta \\ \text{antagonistic} & \text{if } h_c < \min(h_1, h_2) - \delta \\ \text{subadditive} & \min(h_1, h_2) - \delta \leq h_c < \max(h_1, h_2) + \delta \end{cases} \tag{66}$$

If the two drug responses have opposite signs $\operatorname{sgn}(h_1)\operatorname{sgn}(h_2) < 0$, we assume $h_1 > 0, h_2 < 0, h_c >= 0$ for notation simplicity. The interactions are defined by

$$\text{interaction} = \begin{cases} \text{synergistic} & \text{if } h_c \geq h_1 + \delta \\ \text{dominant} & \text{if } h1 - \delta \leq h_c < h_1 + \delta \\ \text{subadditive} & 0 \leq h_c < h_1 - \delta \end{cases} \tag{67}$$

If the two drug responses both have sign 0, the interactions are defined by

$$\text{interaction} = \begin{cases} \text{subadditive} & \text{if } \operatorname{sgn}(h_c) = 0 \\ \text{synergistic} & \text{if } \operatorname{sgn}(h_c) \neq 0 \end{cases} \tag{68}$$

Lastly, if only one of the drug responses has sign 0, we assume $h_1 > 0, \mathrm{sgn}\,(h_2) = 0$ for notation simplicity. The interactions are defined by

$$\text{interaction} = \begin{cases} \text{dominance} & \text{if } \mathrm{sgn}\,(h_c) = 0 \\ \text{synergistic} & \text{if } h_c \geq h_1 + \delta \\ \text{antagonistic} & \text{if } h_c < h1 - \delta \\ \text{subadditive} & \text{if } h1 - \delta \leq h_c < h_1 + \delta \end{cases} \tag{69}$$

## 2.18 Coherence between drug responses

To evaluate the similarity in responses induced by the two drugs, we analyzed both global coherence and program-level coherence. The global coherence was defined by the cosine similarity between the relative response vector of the two drugs, that coherence $= \tilde{\boldsymbol{h}}_1^\intercal \cdot \tilde{\boldsymbol{h}}_2 / \|\tilde{\boldsymbol{h}}_1\| \|\tilde{\boldsymbol{h}}_2\|$. The program-level coherence was determined by comparing the sign of perturbation effect on each gene program. If the relative response vectors of the two single drugs share the same sign on a gene program, the two drugs are coherent on the program. Conversely, opposite signs indicate they are incoherent on the program.

## 2.19 Phase diagrams of dosages of drug combination

The regulatory network of the D-SPIN model is able to convert response vectors into distributions of cell states, therefore enabling the interpolation of cell state transitions between experimental conditions. In the dosage combination experiments with Dasatinib and Halcinonide, we observed the transition between activated macrophage states, monocyte-like resting states, and combinatorial states with both M2 and monocyte transcriptional programs. We used D-SPIN to depict the transitions between the different cell states induced by the change in drug combination dosages.

To precisely tailor the regulatory networks controlling cell state changes in drug dosage combinations, we exclusively used samples from two drug dosage combination experiments to fit a regulatory network $\boldsymbol{J}$ and response vectors $\boldsymbol{h}$. These experiments, including Dasatinib and Halcinonide in one and Tacrolimus and Motolimod in the other, spanned a range of dosages and provided a rich dataset to explore the effect of drug dosage combinations. The first experiment included varying concentrations of Dasatinib and Halcinonide in the following dosages: Dasatinib at 0.1, 1, 10, 100, and 1000nM, and Halcinonide at 0.01, 0.1, 1, 10, 100, and 1000nM. The second experiment included a similar dosage design, using Tacrolimus (dosages of 0.1, 1, 10, 100, and 1000nM), and Motolimod (dosages of 1, 10, 100, 1000, and 10000nM). In both experiments, we profiled all combinations of these two drug dosages.

To elucidate the modulation of cell states with drug combinations, we constructed a phase diagram of cell states by estimating the average log-likelihood of major cell subtypes under every drug dosage combination. For any given drug combination concentration $(c_1, c_2)$, we interpolated the response vector $\hat{\boldsymbol{h}}(c_1, c_2)$ (MATLAB fit 'thinplateinterp') and evaluated the score $u$ of the target cell subtype as the average log-likelihood of individual states $\{\boldsymbol{s}\}_{\text{Target}}$ as

$$u_{\text{Target}} = \frac{1}{\#\{\boldsymbol{s}\}_{\text{Target}}} \sum_{\{\boldsymbol{s}\}_{\text{Target}}} \log P(\boldsymbol{s}; \boldsymbol{J}, \hat{\boldsymbol{h}}(c_1, c_2)) \tag{70}$$

Then, we used softmax across all target cell subtypes to compute the estimated likelihood of each state

$$\hat{p}_{\text{Target i}} = \frac{\exp u_{\text{Target i}}}{\sum_i \exp u_{\text{Target i}}} \tag{71}$$

and rendered the phase diagram by the target state with the largest probability as presented in Figures S7D and S7E.

In the analysis of Dasatinib and Halcinonide combinations, we distinguished five myeloid states of interest, including (a) Activated macrophage: activated control, (b) Monocyte-like: resting control, (c) M2 macrophage: 1000nM Halcinonide, (d) Inhibited monocyte: 1000nM Dasatinib, and (e) M2-like monocyte: 1000nM Dasatinib plus 1000nM Halcinonide. Similarly, in the analysis of Tacrolimus and Motolimod combinations, we investigated three target myeloid states: (a) Activated macrophage: activated control, (b) Monocyte-like: 1000nM Tacrolimus, and (c) Pathogen-response macrophage: 10000nM Motolimod.

# References


[1] Brad T Sherman, Ming Hao, Ju Qiu, Xiaoli Jiao, Michael W Baseler, H Clifford Lane, Tomozumi Imamichi, and Weizhong Chang. David: a web server for functional enrichment analysis and functional annotation of gene lists (2021 update). *Nucleic acids research*, 50(W1):W216–W221, 2022.

[2] Maxim V Kuleshov, Matthew R Jones, Andrew D Rouillard, Nicolas F Fernandez, Qiaonan Duan, Zichen Wang, Simon Koplev, Sherry L Jenkins, Kathleen M Jagodnik, Alexander Lachmann, et al. Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. *Nucleic acids research*, 44(W1):W90–W97, 2016.

[3] David Van Dijk, Roshan Sharma, Juozas Nainys, Kristina Yim, Pooja Kathail, Ambrose J Carr, Cassandra Burdziak, Kevin R Moon, Christine L Chaffer, Diwakar Pattabiraman, et al. Recovering gene interactions from single-cell data using data diffusion. *Cell*, 174(3):716–729, 2018.

[4] Madalina Giurgiu, Julian Reinhard, Barbara Brauner, Irmtraud Dunger-Kaltenbach, Gisela Fobo, Goar Frishman, Corinna Montrone, and Andreas Ruepp. Corum: the comprehensive resource of mammalian protein complexes—2019. *Nucleic acids research*, 47(D1):D559–D563, 2019.

[5] Christopher S McGinnis, David M Patterson, Juliane Winkler, Daniel N Conrad, Marco Y Hein, Vasudha Srivastava, Jennifer L Hu, Lyndsay M Murrow, Jonathan S Weissman, Zena Werb, et al. Multi-seq: sample multiplexing for single-cell rna sequencing using lipid-tagged indices. *Nature methods*, 16(7):619–626, 2019.

[6] Jialong Jiang, David A Sivak, and Matt Thomson. Active learning of spin network models. *arXiv preprint arXiv:1903.10474*, 2019.

[7] H Chau Nguyen, Riccardo Zecchina, and Johannes Berg. Inverse statistical problems: from the inverse ising problem to data science. *Advances in Physics*, 66(3):197–261, 2017.

[8] Pradeep Ravikumar, Martin J Wainwright, and John D Lafferty. High-dimensional ising model selection using $\ell_1$-regularized logistic regression. *The Annals of Statistics*, pages 1287–1319, 2010.

[9] Narayana P Santhanam and Martin J Wainwright. Information-theoretic limits of selecting binary graphical models in high dimensions. *IEEE Transactions on Information Theory*, 58(7):4117–4134, 2012.

[10] Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):192–225, 1974.

[11] Erik Aurell and Magnus Ekeberg. Inverse ising inference using all the data. *Physical review letters*, 108(9):090201, 2012.

[12] Jan Krumsiek, Carsten Marr, Timm Schroeder, and Fabian J Theis. Hierarchical differentiation of myeloid progenitors is encoded in the transcription factor network. *PloS one*, 6(8):e22649, 2011.

[13] Judea Pearl. Reverend bayes on inference engines: A distributed hierarchical approach. In *Probabilistic and Causal Inference: The Works of Judea Pearl*, pages 129–138. 2022.

[14] Judea Pearl. Evidential reasoning using stochastic simulation of causal models. *Artificial intelligence*, 32(2):245–257, 1987.

[15] Barry C Arnold and S James Press. Compatible conditional distributions. *Journal of the American Statistical Association*, 84(405):152–156, 1989.

[16] Sisi Chen, Paul Rivaud, Jong H Park, Tiffany Tsou, Emeric Charles, John R Haliburton, Flavia Pichiorri, and Matt Thomson. Dissecting heterogeneous cell populations across drug and disease conditions with popalign. *Proceedings of the National Academy of Sciences*, 117(46):28784–28794, 2020.

[17] Cameron Sokolik, Yanxia Liu, David Bauer, Jade McPherson, Michael Broeker, Graham Heimberg, Lei S Qi, David A Sivak, and Matt Thomson. Transcription factor competition allows embryonic stem cells to distinguish authentic signals from noise. *Cell systems*, 1(2):117–129, 2015.

[18] Seungjin Choi. Algorithms for orthogonal nonnegative matrix factorization. In *2008 ieee international joint conference on neural networks (ieee world congress on computational intelligence)*, pages 1828–1832. IEEE, 2008.

[19] Stephen A Vavasis. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, 20(3):1364–1377, 2010.

[20] Damian Szklarczyk, Annika L Gable, Katerina C Nastou, David Lyon, Rebecca Kirsch, Sampo Pyysalo, Nadezhda T Doncheva, Marc Legeay, Tao Fang, Peer Bork, et al. The string database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic acids research*, 49(D1):D605–D612, 2021.

[21] Dana Pe er, Aviv Regev, Gal Elidan, and Nir Friedman. Inferring subnetworks from perturbed expression profiles. *BIOINFORMATICS-OXFORD-*, 17:S215–S224, 2001.

[22] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[23] Lisa Holthaus, Daniel Lamp, Anita Gavrisan, Virag Sharma, Anette-Gabriele Ziegler, Martin Jastroch, and Ezio Bonifacio. Cd4+ t cell activation, function, and metabolism are inhibited by low concentrations of dmso. *Journal of immunological methods*, 463:54–60, 2018.

[24] 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, et al. Mapping information-rich genotype-phenotype landscapes with genome-scale perturb-seq. *Cell*, 185(14):2559–2575, 2022.

[25] Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058, 2018.

[26] Aditya Pratapa, Amogh P Jalihal, Jeffrey N Law, Aditya Bharadwaj, and TM Murali. Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nature methods*, 17(2):147–154, 2020.

[27] Luz Garcia-Alonso, Christian H Holland, Mahmoud M Ibrahim, Denes Turei, and Julio Saez-Rodriguez. Benchmark and integration of resources for the estimation of human transcription factor activities. *Genome research*, 29(8):1363–1375, 2019.

[28] Zhi-Ping Liu, Canglin Wu, Hongyu Miao, and Hulin Wu. Regnetwork: an integrated database of transcriptional and post-transcriptional regulatory networks in human and mouse. *Database*, 2015:bav095, 2015.

[29] Heonjong Han, Jae-Won Cho, Sangyoung Lee, Ayoung Yun, Hyojin Kim, Dasom Bae, Sunmo Yang, Chan Yeong Kim, Muyoung Lee, Eunbeen Kim, et al. Trrust v2: an expanded reference database of human and mouse transcriptional regulatory interactions. *Nucleic acids research*, 46(D1):D380–D386, 2018.

[30] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):5233, 2019.

[31] Ctibor Skuta, Martin Popr, Tomas Muller, Jindrich Jindrich, Michal Kahle, David Sedlak, Daniel Svozil, and Petr Bartunek. Probes & drugs portal: an interactive, open data resource for chemical biology. *Nature methods*, 14(8):759–760, 2017.

[32] David S Wishart, Craig Knox, An Chi Guo, Savita Shrivastava, Murtaza Hassanali, Paul Stothard, Zhan Chang, and Jennifer Woolsey. Drugbank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic acids research*, 34(suppl_1):D668–D672, 2006.