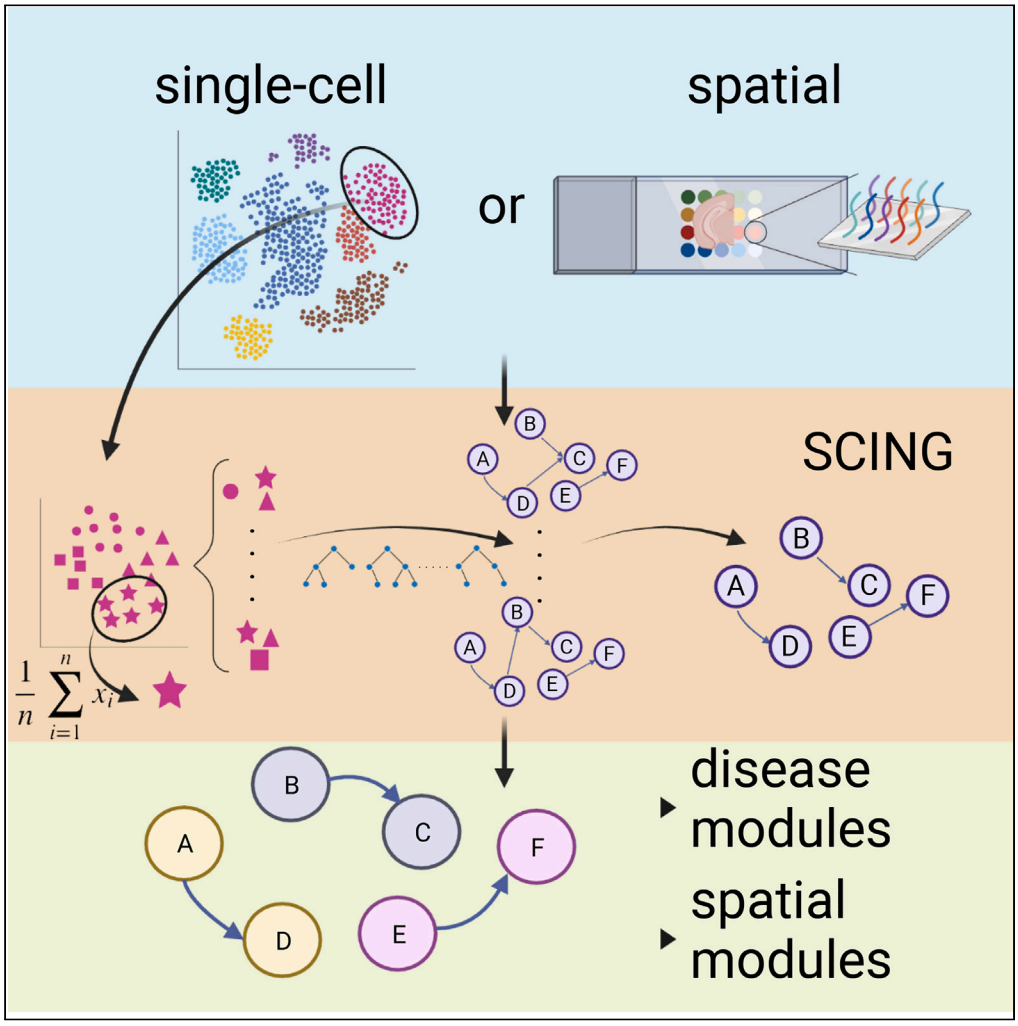
iScience

CellPress
OPEN ACCESS

Article

SCING: Inference of robust, interpretable gene regulatory networks from single cell and spatial transcriptomics



Russell Littman, Michael Cheng, Ning Wang, Chao Peng, Xia Yang

xyang123@ucla.edu

Highlights

SCING infers gene regulatory networks (GRNs) from scRNA-seq or spatial transcriptomics

Perturb-seq and other benchmarking measures highlight SCING GRN accuracy

Application of SCING to Mouse Cell Atlas produced GRNs for 106 cell types

Application of SCING yielded spatial and cell type mechanisms of Alzheimer's disease

Littman et al., iScience 26, 107124
July 21, 2023
https://doi.org/10.1016/j.isci.2023.107124

iScience

CellPress
OPEN ACCESS

Article

SCING: Inference of robust, interpretable gene regulatory networks from single cell and spatial transcriptomics

Russell Littman,[1,2,7] Michael Cheng,[1,2] Ning Wang,[1] Chao Peng,[3] and Xia Yang[1,2,4,5,6,8,*]

SUMMARY

Gene regulatory network (GRN) inference is an integral part of understanding physiology and disease. Single cell/nuclei RNA-seq (scRNA-seq/snRNA-seq) data has been used to elucidate cell-type GRNs; however, the accuracy and speed of current scRNAseq-based GRN approaches are suboptimal. Here, we present Single Cell INtegrative Gene regulatory network inference (SCING), a gradient boosting and mutual information-based approach for identifying robust GRNs from scRNA-seq, snRNA-seq, and spatial transcriptomics data. Performance evaluation using Perturb-seq datasets, held-out data, and the mouse cell atlas combined with the DisGeNET database demonstrates the improved accuracy and biological interpretability of SCING compared to existing methods. We applied SCING to the entire mouse single cell atlas, human Alzheimer's disease (AD), and mouse AD spatial transcriptomics. SCING GRNs reveal unique disease subnetwork modeling capabilities, have intrinsic capacity to correct for batch effects, retrieve disease relevant genes and pathways, and are informative on spatial specificity of disease pathogenesis.

INTRODUCTION

Understanding pathophysiology is necessary for the diagnosis and treatment of complex diseases, which involve the perturbation of hundreds or thousands of genes.[1–4] Identifying perturbed gene pathways and key drivers of complex diseases requires the elucidation of gene regulatory networks (GRN) from high dimensional omics data.[5,6] Previous approaches have been developed and applied to identify these GRNs through bulk transcriptomic data and to determine causal mechanisms of disease.[7–9] More recently, with the advent of single cell RNA sequencing (scRNA-seq) and spatial transcriptomics, the contributions of numerous genes in individual cell types have been implicated in diseases across many disciplines of biology and medicine.[10–12]

GRN construction from scRNA-seq data has been tackled with limited success.[13–15] Existing GRN tools utilize scRNA-seq data with thousands of pre-select genes and cells, because GRNs from full transcriptomes in large scRNA-seq datasets are often computationally intensive and intractable.[13,15] In addition, benchmarking studies have shown limited accuracy of existing methods on both synthetic and real data.[13] The basis of poor performance lies in the technical variability in scRNA-seq data, namely high gene sparsity and cell-to-cell heterogeneity, which bulk GRN methods are not optimized to mitigate and single cell GRN methods are designed to overcome. Top-performing single cell GRN methods based on a recent benchmark study like ppcor,[16] PIDC,[17] and GRNBOOST2,[18] present diverse approaches to GRN construction and unique advantages and limitations that collectively describe the current state of single cell GRN methods. Although tools such as ppcor[16] and PIDC[17] use partial correlation and partial information decomposition, respectively, to identify gene co-expression modules, few methods are able to identify directed networks.[14] Methods that use ensemble machine learning approaches to train GRNs enable the modeling of directed networks. GENIE3 which uses random forest and GRNBOOST2[18] which uses a gradient boosting approach, top performing methods in recent benchmarking studies, with GRNBOOST2 showing superior computing efficiency than GENIE3 and better handling of scRNA-seq dropouts.[13,15] However, both methods generate highly dense networks and require immense computational resources. In addition, most of the regulatory edges point in both directions, and the resulting GRNs contain too many edges in the range of 34,000 to 47,000 edges for networks with only 3,000 input genes for GRNBOOST2, making

[1]Department of Integrative Biology & Physiology, UCLA, Los Angeles, CA, USA

[2]Bioinformatics Interdepartmental Program, UCLA, Los Angeles, CA, USA

[3]Department of Neurology, David Geffen School of Medicine, UCLA, Los Angeles, CA, USA

[4]Institute for Quantitative and Computational Biosciences (QCBio), Los Angeles, CA, USA

[5]Molecular Biology Institute (MBI), Los Angeles, CA, USA

[6]Brain Research Institute (BRI), Los Angeles, CA, USA

[7]Present address: gRED Genentech, 1 DNA way, South San Francisco, CA, USA

[8]Lead contact

*Correspondence: xyang123@ucla.edu

https://doi.org/10.1016/j.isci.2023.107124

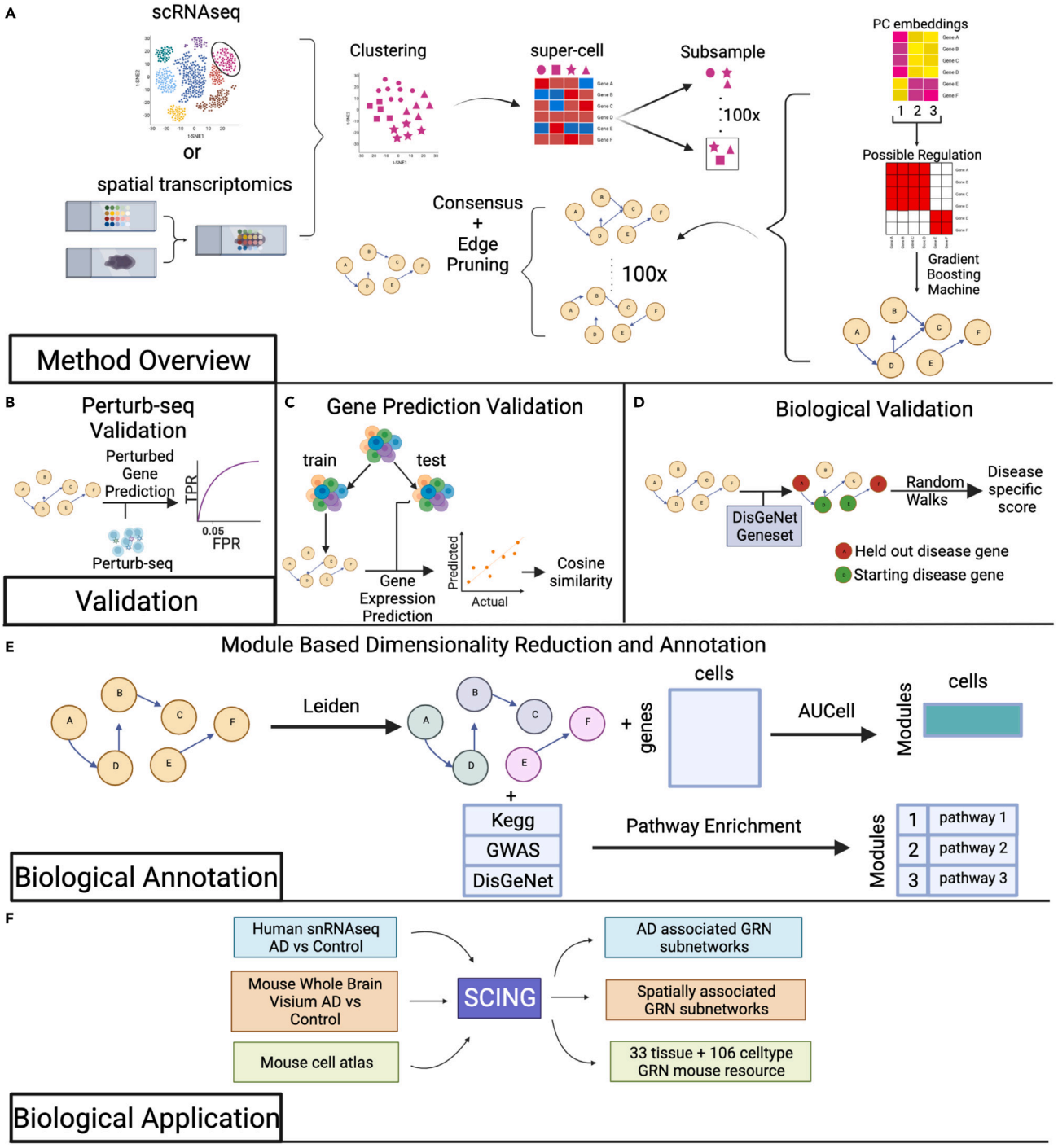


Figure 1. SCING overview, benchmarking, and application

(A) SCING overview. First, we select a specific cell type, or use spatial transcriptomics data. We then cluster the cells/spatial spots using the Leiden graph partitioning algorithm and merge subclusters into supercells. We utilize bagging through subsamples of supercells to keep robust edges in the final GRN. For each subsample, the genes are clustered based on their PC embeddings to limit likely regulatory edges. We then identify edges through gradient boosting regressors (GBR). We find the consensus as edges that show up in 20% of the subsample networks as the default setting, but this threshold can be tuned. We then prune edges and cycles using conditional mutual information metrics.

(B) In silico performance testing using perturb-seq. We identify downstream perturbed genes of each guide RNA targeting a specific gene. We then predict perturbed genes at each depth in the network from the perturbed gene. True positive rate (TPR) and false positive rate (FPR) are determined at each depth in the network. We then utilize AUROC and TPR at FPR 0.05 as metrics for evaluation.

Figure 1. *Continued*

(C) Gene prediction validation and network overfitting assessment. We split data into training and test sets and build a network on the training set. A gradient boosting regressor is trained for each gene based on its parents in the training data. We then predict the expression of each gene in the test set and determine the distance from the true expression through cosine similarity.

(D) Biological validation through disease subnetwork modeling. We utilize a random walk framework from Huang et al. to determine the increase in performance of a GRN to model disease subnetworks versus random GRNs with similar node attributes.

(E) Partition of GRNs into modules and functional annotation of modules. We apply the Leiden graph partitioning algorithm to identify GRN subnetworks and then calculate module specific expression for each cell using AUCell and further combine the gene modules with pathway knowledge bases to annotate modules with biological pathways.

(F) Biological applications of SCING to Alzheimer's disease (AD) datasets and the Mouse Cell Atlas datasets. We apply SCING to human prefrontal cortex snRNA-seq data with AD and Control patients, whole brain Visium spatial transcriptomics data for AD vs. WT mice at different ages, and to the Mouse Cell Atlas for 33 tissues and 106 cell types.

this approach impractical for datasets with thousands of cells and full transcriptomes.[18] SCENIC,[19] an extension of GRNBOOST2, prunes edges based on known transcription factor binding sites (TFBS). However, it only focuses on regulatory behavior between transcription factors (TF) and its downstream target genes,[19] thereby missing other non-TF gene regulatory mechanisms,[20,21] as well as regulatory patterns because of low TF expression.[22] Although there exist other GRN inference approaches based on pseudotime analysis,[23–25] their performance is generally inferior to those that rely solely on the scRNA-seq data.[13]

Here, we present Single Cell INtegrative gene regulatory network inference (SCING), a gradient boosting based approach to efficiently identify GRNs in full single cell transcriptomes. The robustness of SCING is achieved via (1) merging and taking consensus of GRNs through bagging and (2) further directing and pruning edges through the use of edge importance and conditional mutual information. SCING GRNs are then partitioned into modules, to compute module specific expression for each cell. These modules can be used for clustering, phenotypic association, and biological annotation through pathway enrichment. We show our approach is both efficient and robust on large scRNA-seq datasets, able to predict perturbed downstream genes of high throughput perturbation experiments, and produces gene subnetworks with biologically meaningful pathway annotations. We evaluate our approach against GRNBOOST2, ppcor, and PIDC through perturbation target prediction in perturb-seq data, goodness of fit, network characteristic metrics, and disease modeling accuracy. Furthermore, we apply SCING to the mouse single cell atlas,[26] snRNA-seq,[27] and spatial transcriptomics[28] datasets to demonstrate its versatility in datatype accommodation and its biological interpretability of high-throughput transcriptomics datasets. Our code and tutorials for running SCING are publicly available at https://github.com/XiaYangLabOrg/SCING.

RESULTS

SCING method and evaluation overview

SCING leverages the power of abundant cell-level transcriptome data from scRNA-seq/snRNA-seq to identify potential directional regulatory patterns between genes. However, single cell transcriptomics data has technical issues, such as high sparsity and low sequencing depth,[29,30] which make the use of traditional linear or correlative approaches challenging. In addition, these datasets often contain tens of thousands of cells each with hundreds to thousands of genes, making identifying GRNs using complex non-linear approaches on full transcriptomes difficult.[13] To address these limitations, we employ a combination of supercell, gene neighborhood based connection pruning, bagging, gradient boosting regression, and conditional mutual information approaches to identify robust regulatory relationships between genes based on single cell transcriptomics data (Figure 1A; STAR Methods). SCING contains four tunable hyperparameters: the number of supercells to reduce sparsity, the number of subsampled networks for bagging, the number of nearest neighbors for feature selection in gradient boosting regression, and the consensus edge overlap threshold when merging subsampled networks. We benchmarked and optimized these hyperparameters to balance computational efficiency, network properties, and robustness of the GRN based on target gene expression prediction (STAR Methods) and provide a description of the benchmark framework and justification for our selection in the Methods.

We evaluated our approach against PIDC, a partial information decomposition approach; ppcor, a partial correlation approach; and GRNBOOST2, a gradient boosting approach. We selected these particular methods for comparison because of their overall better performance in recent benchmarking studies[13,15] and diverse approaches. We compared these methods on the ability to predict downstream gene targets

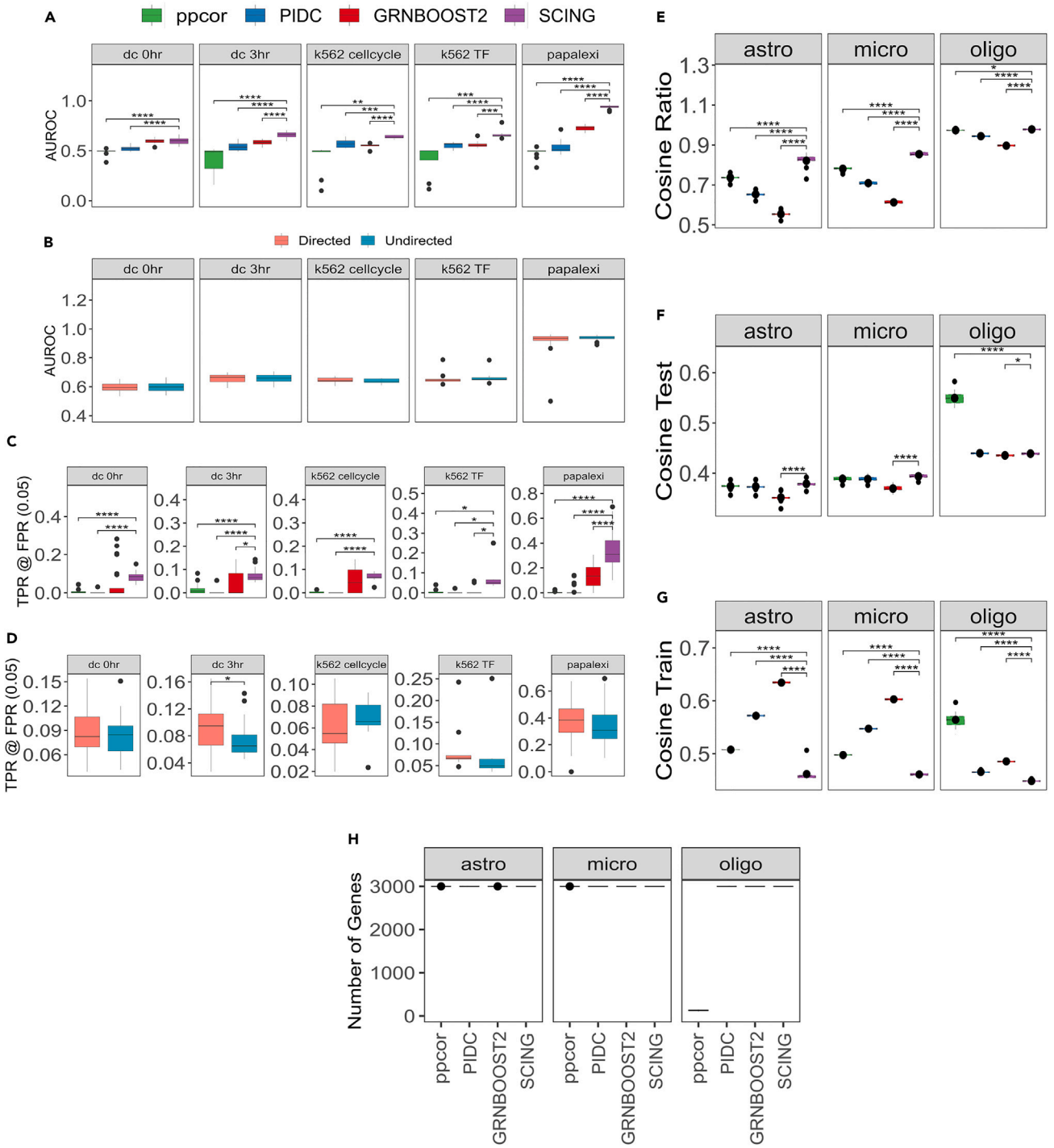


Figure 2. Performance evaluation

(A) Predicted downstream affected genes of perturb-seq based perturbation in 5 datasets with GRNs built on cells with non-zero expression of the perturbation of interest. Area under receiver operator characteristic (AUROC) curve for prediction of downstream perturbations using undirected GRNs.

(B) AUROC for prediction of downstream perturbation on directed GRNs for SCING.

(C) True positive rate (TPR) at a false positive rate (FPR) of 0.05 for the prediction of downstream perturbations on undirected GRNs.

(D) TPR at FPR of 0.05 for the prediction of downstream perturbations on directed GRNs for SCING.

(E) Measure of network overfitting by ratio of cosine similarity between predicted gene expression and actual for testing and training data in held out data for astrocytes, microglia, and oligodendrocytes.

Figure 2. *Continued*

(F) Cosine similarity of predicted gene expression and actual expression in testing data show few differences between SCING and others.

(G) Cosine similarity of predicted gene expression and actual expression in training data shows other methods overfit to the training data.

(H) Gene numbers captured in the cell type networks from each method. p-values between SCING and each of the other methods was computed with an unpaired t-test. (*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$, ****: $p < 0.0001$ unpaired t-test).

of large scale Perturb-seq studies (Figure 1B), robustness of the network on training and test data (Figure 1C), metrics including the consistency of edge overlap on GRNs built on independent cells, and the ability to model disease subnetworks (Figure 1D). Furthermore, we demonstrated the utility of using SCING on the full mouse cell atlas and a human prefrontal-cortex snRNA-seq dataset[27] with AD and control patients to perform snRNA-seq batch harmonization, gene module identification with biological annotation (Figure 1E), and module-trait association analysis (Figure 1F). Data from Morabito et al. has snRNA-seq from 11 AD and 7 control human prefrontal cortex samples, with 61,472 nuclei across 7 cell types, which provides a high-quality dataset for benchmarking. Furthermore, we applied SCING to a visium mouse dataset with AD vs. control samples.[28] We show that the SCING subnetworks are versatile in data type accommodation (scRNA-seq, snRNA-seq, spatial transcriptomics), can resolve spatial biology, and are powerful in retrieving biologically meaningful pathways, gene connections, and disease associations.

SCING extends network node inclusion capacity and improves computing speed

SCING builds many GRNs for each dataset and the speed of such computation is paramount to reasonable computation for a whole dataset. The use of supercells and gene covariance based potential edge pruning enables faster performance of SCING. We show that SCING improves computational speed over GRNBOOST2 and PIDC, when increasing the number of genes (Figure S1A), and number of cells (Figure S1B). GRNBOOST2 scales exponentially on the number of genes, whereas PIDC scales exponentially on the number of cells, making whole transcriptome and large dataset GRN inference difficult. Supercells in SCING ensure the network building run time does not increase as a function of cells and potential edge pruning enables linear increase in computation with respect to genes. We note that ppcor's fast general matrix formulation improves GRN inference time compared to all other approaches, including SCING. Although SCING is slower than ppcor, it performs inference on 4,000 genes in ~21 s for all cell types, which is reasonable to compute hundreds of GRNs for any given sample.

SCING GRNs better predict downstream genes of perturbed genes in perturb-seq

We tested whether GRNs from each approach can predict gene expression changes in downstream genes from gene knockdown treatments. Here, we used Perturb-seq datasets, which enabled us to identify the effects of many perturbed genes in parallel. We utilize previously published datasets with THP-1, dendritic (DC), and K562 cells with 25, 24, and 21 genes perturbed, respectively.[31,32] The DC cells were split into lipopolysaccharide (LPS) stimulated and non-stimulated cells with perturbations targeting transcription factors (TFs) as two datasets, and the K562 cells were split into two datasets based on the genes initially perturbed (TFs or cell cycle related genes) in the Perturb-seq experiments. THP-1 cells contain perturbations targeting PD-L1 regulators.

We identified genes downstream of each perturbation through an elastic net regression framework to determine the effect of RNA guides on each gene while regressing out cell state[32] (STAR Methods) (Table S1). We compared GRNs generated from SCING to GRNs generated from GRNBOOST2, PIDC, and ppcor in predicting genes downstream of each target gene in each perturb-seq experiment. For any given network, we iterated through the downstream genes of an initially perturbed gene by the RNA guide and determined if the predicted downstream genes were significantly altered. We determined the true positive rate (TPR) and false positive rate (FPR) at each network depth to compute the area under the receiver operating characteristic (AUROC) curve, as well as the TPR at an FPR of 0.05. We examine TPR at FPR 0.05 to show perturbation prediction accuracy in a setting more relevant to biological analysis (controlling for FPR 0.05). We first examine the prediction performance of each GRN approach when building GRNs on datasets with cells removed that have zero expression of the target gene. Removing these cells mitigates performance effects from sparsity (Figure S2A). Since ppcor and PIDC produce undirected graphs, and GRNBOOST2 generally has bidirectional edges, we first evaluated SCING against the other methods without considering edge direction, showing a higher AUROC for SCING (Figure 2A). Edge direction does not affect this metric (Figure 2B). In addition, we show SCING improves TPR at FPR of 0.05 (Figure 2C). Edge direction only affects prediction accuracy in TPR at FPR 0.05 for dc 3h cells (Figure 2D).

Across all Perturb-seq datasets tested and when no cells were removed, SCING either outperformed, or met the performance of the other tools in both AUROC (Figure S2B) and TPR at FPR 0.05 (Figure S2D) and performance was minimally affected by edge direction (Figures S2C and S2E). For AUROC, SCING outperforms ppcor across all datasets, PIDC across 4 datasets, and GRNBOOST2 across 2 datasets. For TPR at FPR 0.05, SCING outperforms ppcor and PIDC across 4 datasets and GRNBOOST2 across all datasets (Figure S2D). SCING performed best in the Papalexi et al. THP-1 dataset, although there is a large variance in both AUROC (Figure S2B) and TPR at FPR 0.05 (Figure S2D) for both ppcor and SCING.

Overall, SCING outperforms all other approaches at predicting perturbation effects in Perturb-seq data when sparsity is adjusted (Figure 2) and outperforms select methods when all cells are used (Figure S2). Thus, we recommend removing cells with sparse gene expression when using SCING.

SCING mitigates overfitting and builds more robust GRNs

Models often overfit to their data and fail to properly perform on new datasets. In the GRN context, we aimed to identify connections and networks that are able to capture biological variation rather than sample or batch specific effects. To test the performance of GRN inference approaches and their ability to capture robust biological signals, we tested the ability of a model trained on parents of each gene in training data to predict the gene expression of the downstream target gene in testing data. We split the scRNA-seq data from control human prefrontal cortex[27] into training and testing sets (STAR Methods). First, we built GRNs on the training data from oligodendrocytes, astrocytes, and microglia using SCING, ppcor, PIDC, and GRNBOOST2. Subsequently, we trained gradient boosting regressors for each gene based on the parents in a given network using the training data. The trained regressors were then used to predict the gene expression of cells in testing data based on the expression of the parent genes in those cells. We evaluated the performance of each GRN approach by averaging the cosine similarity score over all downstream genes that have parents in the network. This process was repeated for 10 replicates on random subsamples of 1,000, 3,000, and 5,000 genes based on runtime feasibility (Figure S3).

To measure overfitting, we used the cosine similarity score ratio between the test and training sets, with a higher ratio indicating lower overfitting. We found that SCING GRNs had less overfitting than the other approaches (Figures 2E and S3A–S3C). In terms of performance in the test sets, SCING performed similarly to ppcor and PIDC and outperformed GRNBOOST2 (Figure 2F). On training data, GRNs from ppcor, PIDC, and GRNBOOST2 had higher cosine similarity scores compared to SCING, reflecting overfitting on the training data by the other methods (Figure 2G). We noted that the number of genes in the resulting network to be very low in the ppcor oligodendrocyte network (Figure 2H), which likely affected the results of ppcor as evaluated by the cosine similarity measure here. In addition, we note that ppcor could not build networks for microglia with 1,000 genes (Figure S3B). These results support the robustness of SCING GRNs with less overfitting.

SCING fits scale free model and shows edge consistency

As another measure of GRN quality and performance, we compared GRNs generated by each method by various standard network metrics (scale-free network fit, number of edges, number of genes, and betweenness centrality), as well as robustness of network edges between networks on 50/50 split datasets. We tested this on 10 replicates for each of the 3 cell types (oligodendrocyte, astrocyte, and microglia) in the scRNA-seq data from control human prefrontal context, with 3,000 different genes randomly selected for each subsample of cells.

GRNs are thought to follow a scale-free network structure, in which there are few nodes with many connections, and many nodes with few connections.[33] We computed scale-free network structure through the R-squared coefficient of a linear regression model regressing on the log of each node's degree and the log of the proportion of nodes with that given degree. We show that the R-squared value for SCING is significantly higher than PIDC and GRNBOOST2 methods and is trending higher than ppcor, indicating SCING networks more closely follow a scale free network structure (Figure 3A). In a typical scale-free network plot of log10 node count versus log10°, we expect a power law distribution. However, we found that networks built on scRNA-seq data have a parabolic distribution, with only the right half following a power law distribution whereas the left portion of the plot is driven by genes that were very sparse and likely do not fit typical distributions (Figure 3B). Therefore, we excluded the sparse genes from the scale free regression calculation based on a sparsity threshold of 0.7.

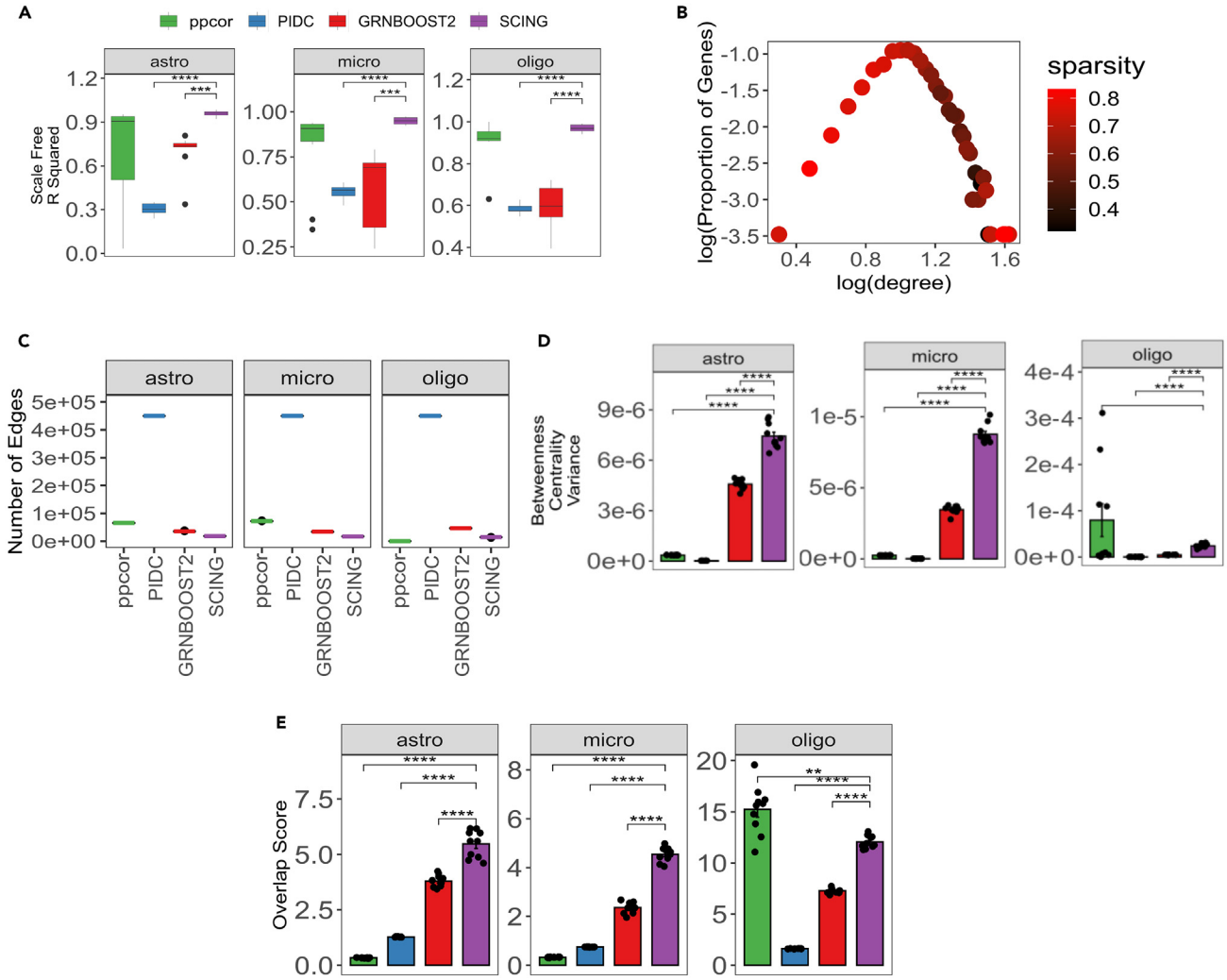


Figure 3. Network features and consistency

(A) Descriptive features of networks across SCING and other approaches for 10 networks on astrocytes, microglia, and oligodendrocytes. Linear regression R-squared for log degree vs. log count for goodness of fit metric of scale-free network.

(B) Example scatterplot of log degree vs. log count with the average sparsity of genes in each dot. Brighter red indicates higher sparsity. This shows highly sparse genes tend to have lower degrees. We excluded points based on a sparsity threshold of 0.7 before computing the scale free regression coefficient.

(C) Average number of edges for each method across all cell types.

(D) The variance of the betweenness centrality across nodes in each graph.

(E) The overlap score (number of overlapping edges/expected number of edges) in independent sets of cells. p-values between methods computed with an unpaired t-test between SCING and each of the other methods. (*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$, ****: $p < 0.0001$). Error bars represent standard error.

Among all four methods tested, PIDC produced the largest networks (Figure 3C), followed by ppcor, GRNBOOST2, and SCING. SCING networks have much fewer edges than the other approaches while keeping similar numbers of genes (Figure 2H) and better performance. Although network size is not a proxy of network accuracy, a smaller network with robust and consistent edges helps reduce overfitting (Figures 2E, 2F, and 3C). One exception is that the ppcor oligodendrocytes network contains only 214.8 edges on average compared to 14,545.8, 46,891.9, and 449,850 in SCING, GRNBOOST2, and PIDC, respectively. Many genes without regulatory edges were not included in the final ppcor network. The smaller ppcor oligodendrocyte network has implications for the betweenness centrality and edge overlap metrics, as follows.

Betweenness centrality is often used as another metric to determine the overall connectedness of a graph.[34] For a given node, betweenness centrality is the number of shortest paths that pass through

that node, indicating how much information that node presents to the graph. High betweenness centrality indicates that a node conveys a lot of information to a given graph. We found that SCING networks generally have higher variance of betweenness for the nodes in the networks (Figure 3D). This indicates that some nodes are more centralized than others when compared to other approaches, again consistent with the scale-free network model.

To determine network consistency, we split each cell type into two groups of non-overlapping cells. We built networks for each dataset using all methods and calculated the fraction of total edges that overlap between the two networks. Although larger networks tend to have more edge overlap, this also holds for larger random networks. We designed a normalized overlap score: the fraction of edges overlapped divided by the expected number of overlapping edges of a random network of the same size (STAR Methods). When controlling for network size, SCING has significantly more overlap, or higher reproducibility, between networks of 50/50 split data than the other approaches (Figure 3E).

SCING more accurately model disease subnetworks

To evaluate the performance of GRNs on disease modeling, we applied an approach developed by Huang et al.[35] Briefly, given a known disease gene set and a GRN, we evaluate the ability of the GRN to reach held out disease genes by starting from select disease genes in the network through random walks. We then compared the performance for each network to that of a random network in which the nodes follow similar degree characteristics to derive a performance gain measurement. Here, we selected known gene sets for 3 classes of diseases from DisGeNET[36] (Immune, Metabolic, and Neuronal) (Table S2) and obtained scRNA-seq data for cell types from 3 tissues relevant to each disease class from the mouse cell atlas[26] (bone marrow for immune diseases, brain for neuronal diseases, and liver for metabolic diseases) (Table S3). First, to reduce the number of genes, we filtered the scRNA-seq data by removing genes expressed in fewer than 5% of cells and added expressed disease genes from all DisGeNET disease gene sets. We built GRNs using each method and evaluated the performance gain over random networks on the disease gene sets. We found that across all tissues and all disease types, SCING outperformed all other approaches (Figure 4A).

Application case 1: Constructing SCING GRNs using mouse cell atlas (MCA) scRNA-seq datasets to interpret diseases

After establishing the performance of SCING GRNs using the various approaches described above, we established an SCING GRN resource for diverse cell types and tested the broader utility of SCING to produce biologically meaningful GRNs. To this end, we applied SCING to generate GRNs for all cell types with at least 100 cells in all tissues of the MCA. We constructed a total of 273 cell-type specific networks, across 33 tissues and 106 cell types. To identify which GRN informs on which disease, we applied the above random walk approach from Huang et al. and summarized the results in (Figures S4A–S4C, Table S4). We found clusters of cell type GRNs defined by DisGeNET diseases that had similar patterns (Figures S4A and 4B). Some disease genes can be modeled well using GRNs from numerous cell types (Figure S4A) whereas others are more cell type or tissue specific (Figure S4B). In addition, some cell type GRNs are able to model a broad range of diseases (Figures S4A and S4C). We found that immune cell type (light blue squares in Figure 4B) GRNs can model a wide range of diseases, whereas non-immune cell type GRNs (light purple squares in Figure 4B) are more specific to vasculature related diseases.

We further explored the dynamics of GRNs of immune cell types across all diseases in DisGeNET. We clustered the cell types in the performance gain matrix with only immune cell types included, and sorted the GRNs by the number of diseases they can accurately model (Figures 4C, 4D, S5A, and S5B). We noticed that cell types of the innate immune system can model a broader range of diseases than those of the adaptive immune system[37] (Figure 4E).

Our SCING cell type GRNs resource and the above patterns of relationships between cell type GRNs and diseases support the utility of the SCING cell type GRN in disease interpretation. The networks can be accessed at https://github.com/XiaYangLabOrg/SCING to facilitate further biological mining of complex diseases.

Application case 2: Using SCING GRNs to interpret Alzheimer's disease (AD)

We next applied SCING to a single nuclei RNA-seq (snRNA-seq) dataset from Morabito et al. that examined human prefrontal cortex samples from AD and control patients to evaluate the applicability of SCING GRNs in understanding AD pathogenesis.[27] We focused on microglia because of their strong

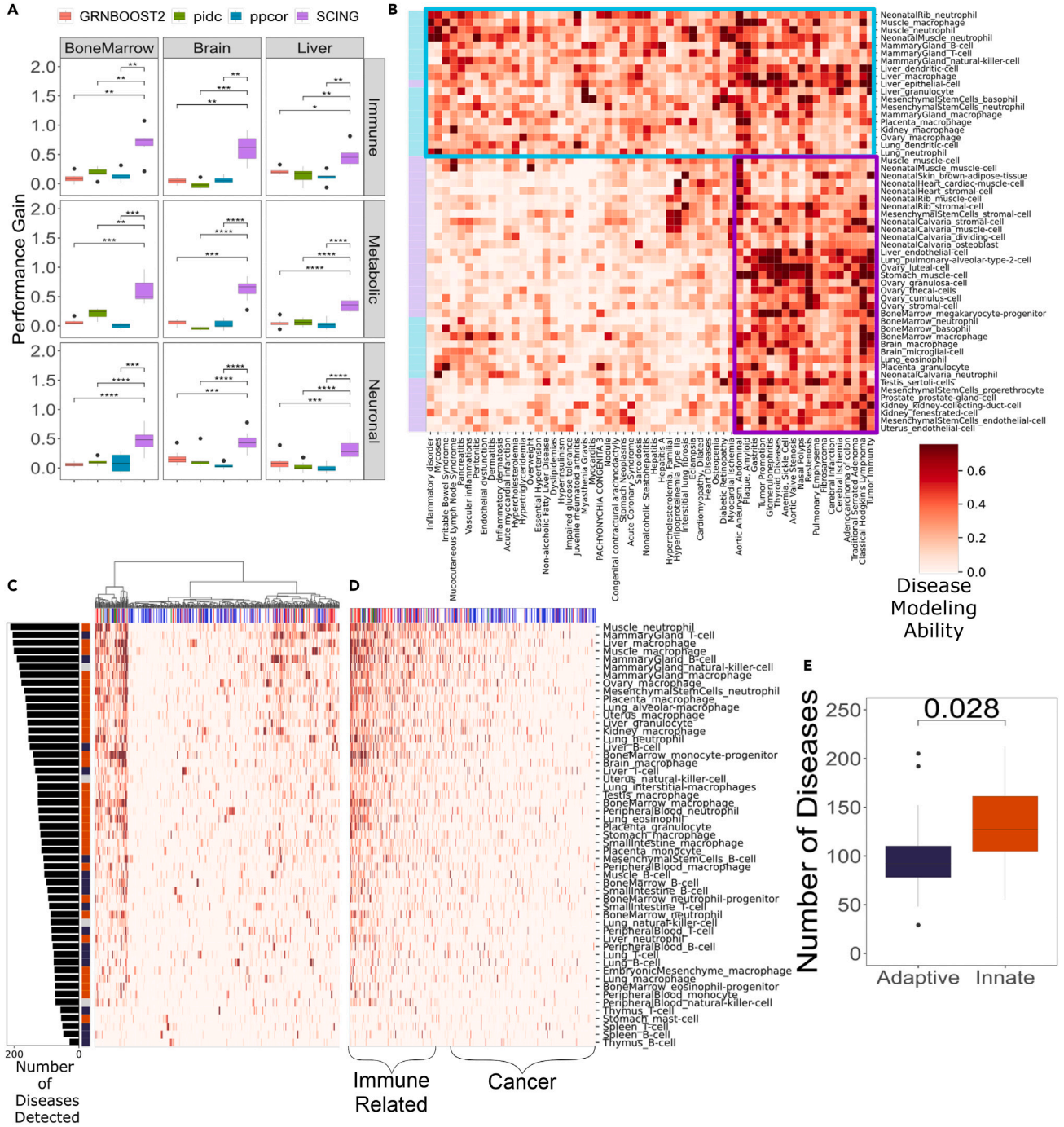


Figure 4. Application case 1: constructing and using SCING GRNs based on Mouse Cell Atlas scRNA-seq datasets to interpret diseases

(A) Performance of modeling disease subnetworks for DisGeNET gene sets related to the immune, metabolic, and neuronal diseases with GRNs built on bone marrow, brain, and liver cells, reveals SCING models disease subnetworks more accurately than other methods.

(B) Clustermap depicting GRNs built with SCING from immune cell types (light blue), model disease subnetworks from many different disease gene sets, whereas vascular cell types (purple) are more specific to vascular diseases. Cell types (rows) from the adaptive (blue) and innate (orange) immune systems, show variability in the number of diseases (columns) they model (>0.1).

(C and D) Clustermap shows diseases clustered with hierarchical clustering and sorted by the number of cell types that can accurately model that disease subnetwork. Diseases are colored by disease category (immune related: red; cardiothoracic: green; cancer: blue; immune related cancer: purple), and cell types are colored by innate (orange), and adaptive immune system (dark blue).

(E) Innate immune system cell types better model disease subnetworks from more diseases. p-values between methods computed with an unpaired t-test. (*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$, ****: $p < 0.0001$).

implication in AD[38,39] and a better current understanding of the genes and biological pathways in microglia in AD, to demonstrate that SCING GRNs can retrieve known biology.

The SCING microglia GRN contained 10,159 genes and 63,056 edges. Using the Leiden clustering algorithm,[40] we partitioned the SCING microglia GRN into 21 network modules. Next, we summarized module-level expression for each cell using the AUCell method from SCENIC on the partitioned GRN modules.[19] When cells were clustered based on the raw gene expression values, as is typical with human samples, cells from individual samples clustered together (Figure 5A), making it difficult to isolate sample heterogeneity from biological variability. However, when using SCING module expression to cluster cells, the sample, batch, and RNA quality effects were mitigated (Figures 5A and 5B). In contrast, biologically relevant variation, such as sex, AD diagnosis, and mitochondrial fraction were better retained (Figures 5C and 5D). In the UMAP control cells tend to localize to the right side, whereas cells from females tend to localize to the top part. These results suggest that SCING GRNs have intrinsic ability to correct for non-biological variations.

To quantitatively evaluate how well SCING GRNs can be used for batch effect correction and biological preservation, we compared SCING GRNs with commonly used batch correction methods, such as FastMNN,[41] Harmony,[42] and Seurat,[43] chosen based on their better performance in previous benchmarking studies.[44] We first performed dimension reduction and clustering of cells based on corrected data from each batch correction method. For SCING, the values used were SCING GRN module AUCell scores. We took each cell, determined how many neighbors in the PC space had the same annotation of interest (sample, batch, diagnosis, etc.), and then scored each batch correction approach by the fraction of cells that had the same annotation. We removed batch and sample specific effects using the F1 score (STAR Methods).[44] We found that the SCING GRN module based dimensionality reduction carried the ability to correct for batch effect and retain biological information (Figure S6) in a similar manner to dedicated batch correction methods such as FastMNN,[41] Harmony,[42] and Seurat.[43] Although SCING GRN based batch correction was not as optimized as the dedicated batch correction methods, many typical batch correction methods do not provide a batch corrected gene expression matrix for downstream analysis.[44] SCING GRN modules have direct biological interpretability without prior batch correction, since each SCING module can be associated with phenotypic traits but not batch and annotated with pathways, as described below. We further note that SCENIC[19] has shown similar robustness to batch effect and that this is likely a general characteristic of GRN-based dimensionality reduction approaches that is not specific to SCING. This general characteristic offers network methods an important advantage over individual gene-based analysis.

We identified SCING GRN modules associated with AD diagnosis, plaque stage, and tangle stage through linear regression of the phenotypic traits and each module's expression across cells, while regressing out sex specific differences. We found that ~43% of modules were significantly (FDR <0.05) associated with at least one trait and ~24% of modules were significantly associated with all three traits. To examine the biological interpretation of these modules, we performed pathway enrichment on the genes in each module utilizing the GO biological process, DisGeNET, Reactome, BioCarta, and KEGG knowledge bases. We found that 78% of the significantly trait-associated modules were significantly enriched for biological pathways (Figure 5E, Table S5). These modules recapitulated pathways related to AD (module 9), immune processes (modules 0, 2, 9, 13, and 19), cytokine triggered gene expression (modules 12, 18, 19), and endocytosis (modules 2, 9, and 13). These are expected perturbed pathways for microglia in AD.[38]

We dug into the vesicle mediated transport pathway from module 2 and visualized the network with the differential gene expression of each gene (Figure S7). This microglia pathway is important in AD.[45] In addition, we found the APOE and APP subnetwork within the vesicle mediated transport pathway which are among the most significant genetic risk factors for AD (Figure 5F).[46–48]

Therefore, our results demonstrate that SCING GRNs can correct for batch effects intrinsic to scRNA-seq studies and can recapitulate known cell type specific genes, pathways, and network connections.

Application case 3: Using SCING to model GRNs based on 10x Genomics Visium spatial transcriptomics data to interpret AD

To evaluate the applicability of SCING beyond scRNA-seq and snRNA-seq, we next applied SCING to spatial transcriptomics data from Chen et al.[28] as a new approach for spatial transcriptomics analysis. We built an SCING GRN on all spots in the visium data to obtain a global GRN with 128,720 edges across

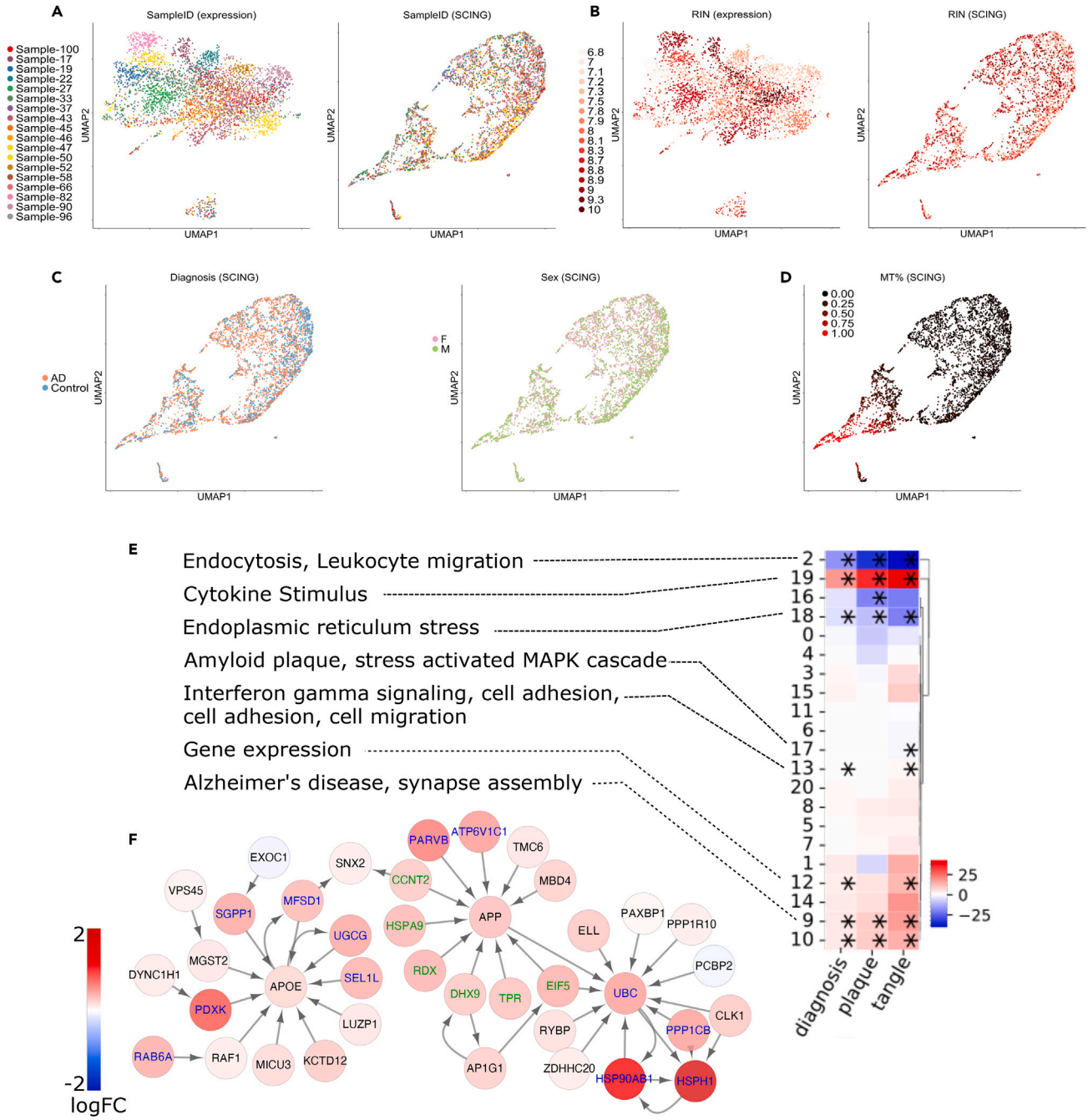


Figure 5. Application case 2: Using SCING GRNs to interpret Alzheimer's disease (AD)

(A) UMAP representation of scRNA-seq data shows sample specific differences when operating on gene expression space (left panel). Dimensionality reduction on SCING module embeddings removes sample specific effects (right panel).

(B) SCING removes RNA quality effects on gene expression clustering.

(C and D) Clustering on SCING modules, keeps biologically relevant features such as AD status (left panel) and sex (right panel), as well as mitochondrial fraction.

(E) Heatmap showing coefficients of linear regression of diagnosis, plaque, and tangle status, on module expression, while regressing out sex. Pathway annotations for significant (*: FDR <0.05) modules are provided.

(F) Subnetwork reveals collocalization of canonical Alzheimer's genes and subnetworks such as APOE and APP.

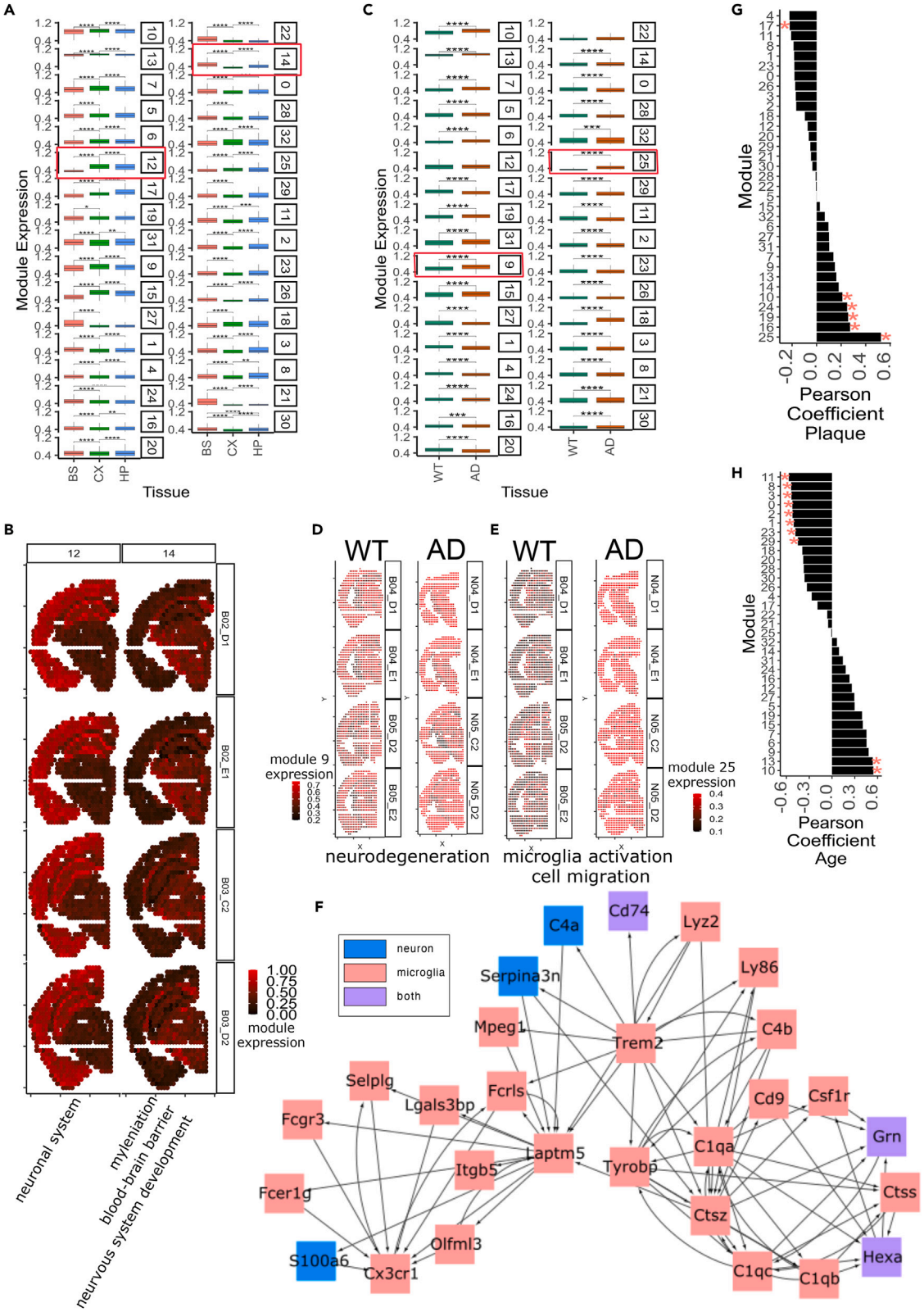

A
Module Expression
Tissue
BS
CX
HP
C
Module Expression
Tissue
WT
AD
G
Module
Pearson
Coefficient
Plaque
B
12
14
B02_D1
B02_E1
B03_C2
B03_D2
neuronal system
myleniation
blood-brain barrier
neurvous system development
module
expression
D
WT
AD
module 9
expression
neurodegeneration
E
WT
AD
module 25
expression
microglia activation
cell migration
H
Module
Pearson
Coefficient
Age
F
neuron
microglia
both
C4a
Cd74
Lyz2
Serpina3n
Ly86
Mpeg1
Trem2
C4b
Selplg
Fcrls
Lgals3bp
Fcgr3
Cd9
Csf1r
Grn
Laptm5
Tyrobp
C1qa
Itgb5
Fcer1g
Ctss
Ctsz
Olfml3
S100a6
Cx3cr1
C1qc
C1qb
Hexa

Figure 6. Application case 3: Using SCING to model GRNs based on 10x Genomics Visium spatial transcriptomics data to interpret AD

(A) Boxplots show regional specificity of specific modules, namely module 12 and 14 (red boxes).

(B) Regional specificity of module 12 (hippocampus and cortex, enriched for neuronal system pathways), and module 14 (hypothalamus, thalamus, and fiber tract, enriched for microglial activation and cell migration pathways), visualized on brain samples of 3-month-old WT mice. Enriched pathways were labeled at the bottom of each module.

(C) Module association in AD versus WT mice. Red boxes indicate modules 9 and 25, which are further visualized.

(D) Visualization of AD-associated module 9 (enriched for neurodegeneration) in 18-month-old WT and AD mice.

(E) Visualization of AD-associated module 25 (enriched for microglia activation and cell migration) in 18-month-old WT and AD mice.

(F) Module 25 subnetwork of Trem2 and complement proteins shows microglial association of module 25. Full subnetwork in Figure S11. Nodes are colored by marker gene status of neuronal (blue), microglial (red), or both (purple), as determined from the Allen Brain Atlas. Cross cell type communication edges seen between red and blue.

(G and H) Pearson correlation between module expression of each module and plaque (G) or age (H). Significance values for each module are determined by a null distribution generated from 1,000 random subsamples of genes with the same gene number of a given module. The AUCell was computed for each random module and Pearson correlation coefficients were calculated. The p value for the true module was computed based on the null distribution of the correlation coefficients, and p values were further corrected for multiple testing using Bonferroni correction. Significance is shown (*: adjusted p value <0.05). Subpanel a,c: (*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$, ****: $p < 0.0001$ by unpaired t-test).

15,432 genes. In addition, Chen et al. profiled beta amyloid plaque with immunohistochemistry, and we included this protein expression value in the SCING GRN, highlighting the possibility of constructing multi-omics networks using SCING.

We partitioned the genes in the resulting SCING GRN with the Leiden graph partition algorithm into 33 modules and performed the AUCell score from SCENIC[19] to obtain module specific expression for each spot and annotated the enriched pathways for each module (Table S6). We found module specific expression in the mouse brain subregions (Figures 6A, 6B, and S8), based on clustering of the average module expression across spots in each brain subregion (Figure S9). We found cortical subregions to cluster together, as well as the thalamus and hypothalamus (Figure S9), based on GRN module expression patterns. We also identified modules more specifically expressed in the cortex and hippocampus (CS, HP) (module 12), or the fiber tract, thalamus, and hypothalamus (BS) (module 14) (Figures 6A and 6B). Module 12 was highly enriched for genes involved in neuronal system, axonogenesis, and chemical synapse, which might reflect the dynamic status of hippocampus and cortex neurons for memory formation and cognitive function. In contrast, module 14 was enriched in genes involved in myelination[49] and blood-brain barrier,[50] consistent with the high enrichment of oligodendrocyte populations in the fiber tracts, and indicated blood-brain barrier changes in the thalamus. We also found modules that separate more similar subregions from one another such as module 27 (enriched for axonogenesis) and 21 (enriched for calcium ion transport) expressed much higher in the thalamus than in the hypothalamus (Figure S9). By contrast, modules, such as module 5 (enriched for chromatin organization and peroxisomal lipid metabolism) and module 19 (enriched for ribosomal biogenesis and protein processing), are much less specific to subregions (Figures S8 and S9). These are general cellular functions that are expected to have broad expression across the brain.

We leveraged the Allen Brain Atlas to determine the marker gene proportion of each SCING module based on the genes in the module. We determined marker genes for each of neurons, microglia/perivascular macrophages, astrocytes, and oligodendrocytes from the Allen Brain Atlas scRNA-seq data. We then determined the proportion and enrichment of genes in each module that are marker genes for each cell type and plotted the data as heat maps (Figure S10). We see that many modules are mostly made up of neuronal genes, however some modules have higher contributions from oligodendrocytes (modules 14, 26), and microglia (modules 25, 16, 19, 20). This indicates that utilizing high quality reference data can help to deconvolve cell type membership of SCING modules.

We found many modules to be AD associated (Figure 6C), in particular, module 9 (enriched for neurodegeneration) (Figure 6D) and module 25 (enriched for microglial activation, lysosome, and cell migration) (Figure 6E). We further explored these subnetworks and found most of the genes in module 9 to be neuronal marker genes, whereas most of the genes in module 25 to be microglial marker genes (Figure S11). The module 25 subnetwork to contain the *Trem2* and *C1q* subnetworks, highly profiled in AD microglial cells (Figure 6F). Of interest, we found a cross-module edge and several cross-cell-type edges, likely revealing intercellular communications.

We identified 6 modules that were significantly associated with amyloid beta plaque (Figure 6G) through Pearson correlation and null distribution from a permutation approach (STAR Methods). Modules 25

(enriched for immune function) and 19 (enriched for nonsense mediated decay, and infectious disease) were among the most highly correlated with plaque and were also associated with AD (Figure 6C).

We found that amyloid plaque staining could be partitioned based on the expression of module 10 in the SCING GRN (Figure S12), which had significantly higher expression in AD mice than in WT (Figure 6C), and is also quantitatively correlated with plaque (Figure 6G). Module 10 contains genes highly related to metabolism, neurodegenerative diseases, and immune function (Table S6). We show the subnetwork for the amyloid beta plaque in module 10 (Figure S12). In addition, module 25 (enriched for microglial activation, lysosome, and cell migration) was also highly associated with plaque (Figure 6G), as expected in AD pathogenesis.

In addition to plaque association, we explored the age-related modules through Pearson correlation between age (6, 12, and 18 months) and module expression in WT samples. We found 10 modules to be age related (Figure 6H). Two highly positively associated modules with age, module 10 (subnetwork containing amyloid protein) and 13, have age-related pathways. Module 10 is involved in protein degradation, neurodegeneration, and cell cycle, and module 13 is involved in metabolism, cellular response to stress, and immune system (Figure 6H, Table S6).

We found certain modules such as module 30 (enriched for neuropeptide signaling) to be spatially variable (Figure S13) and associated with AD (Figure 6B). Based on the locations from the Allen Brain Atlas[51] (Figure S13A), we find module 30 to be more specific to the hypothalamus than other regions of the BS (Figure S13B). We also find that within the hypothalamus, module 30 expression is higher in 18-month-old AD mice compared to the WT mice (Figure S13C). Hypothalamic alterations have been observed in the hypothalamus in AD development.[52] The module 30 subnetwork (Figure S13D) shows key hypothalamic neuropeptides, such as *Pomc*[53] and *Pnoc*[54] which are consistent with their enrichment in the hypothalamus.

Our applications of SCING to spatial transcriptomics data demonstrate its broader utility beyond scRNA-seq/snRNA-seq and revealed spatial network patterns of AD.

DISCUSSION

Single cell multiomics has become a powerful tool for identifying regulatory interactions between genes, but the performance of existing tools is limited in both accuracy and scalability.[13] Here, we present SCING, a gradient boosting, bagging, and conditional mutual information based approach for efficiently extracting robust GRNs on full transcriptomes for individual cell types. We validate our networks using a novel perturb-seq based approach, held-out data prediction, and established network characteristic metrics (network size, network overlap, scale-free network, and betweenness centrality) to determine performance and network features against other existing tools. SCING not only offers robust and accurate GRN inference and improved gene coverage and speed compared to previous approaches,[9,18] but also versatile GRN inference with scRNA-seq, snRNA-seq, and spatial transcriptomics data. Using various application examples, we show that SCING infers robust GRNs that inform on cell type specific genes and pathways underlying pathophysiology while simultaneously removing non-biological signals from data quality, sample, and batch effects through gene regulatory module detection and functional annotation. We also provide a comprehensive SCING GRN resource for 106 cell types across 33 tissues using data from MCA to facilitate future applications of single cell GRNs in our understanding of pathophysiology.

SCING efficiently identifies robust networks using supercells, a bagging approach, and mutual information-based edge pruning, to remove redundant edges in the network. The supercell and reduction in potential edges make the bagging approach possible by removing computational time for each GRN. GRNBOOST2 uses a similar framework as SCING (gradient boosting regression) but overfits the data by generating too many edges to be interpretable, which are also undirected, making them less useful for biological interpretation. Meanwhile, ppcor and PIDC use partial correlation and partial information decomposition approaches, which are more accurately described as measures of coexpression, rather than gene regulation. In contrast, the directed graphs built with SCING show better perturbation prediction and consistency across replicates.

Validation of GRN inference tools has remained challenging.[13] Since ground truth networks are unknown for many conditions across cell types and tissues, current GRN accuracy evaluations rely on inferring synthetic ground truth networks that may not capture biological or curated interactions that forsake tissue and cell-type specificity.[13] Our novel Perturb-seq based approach provides a unique way to determine GRN accuracy,

since it leverages cells with specific gene perturbations to infer downstream genes. Prediction of perturbed genes is a very powerful aspect of GRN construction, and SCING stands out above all other methods in this regard.[55] This is the key difference between our benchmarking framework and that used in BEELINE by Pratapa et al., whereas the other aspects of the benchmarking framework are conceptually similar.

We demonstrate that SCING networks are applicable to scRNA-seq, snRNA-seq, and spatial transcriptomics data. Our network-based module expression is robust to batch effects and provides biologically annotated expression values for each cell that can be directly used for disease modeling (Figure 4), phenotypic correlation (Figure 5), and further spatial analysis (Figure 6) to boost our ability to interpret single cell data. We note that SCING for spatial transcriptomics network analysis does not currently utilize the spatial information during network construction, although the resulting network modules intrinsically carry a certain level of spatial information (Figure 6). Explicitly leveraging the spatial information will likely enhance network performance and is an important future direction to improve SCING for network modeling of spatial transcriptomics data.

At the intersection of single cell omics and complex diseases, SCING provides sparse but robust, directional, and interpretable GRN models for understanding biological systems and how they change through pathogenesis. GRNs can be analyzed to identify and predict perturbed subnetworks, and as a result, be used to investigate key drivers of disease.[56] Identifying key drivers of disease by teasing apart biology and technical variation from high throughput, high dimensional datasets will lead to more successful drug and perturbation target identification, as well as robust drug development.[57–59]

We note that SCING is currently tested to infer GRNs based on individual scRNA-seq, snRNA-seq, and spatial transcriptomics data. Other types of omic information such as scATAC-seq, scHi-C, or cell type specific *trans*-eQTL information can be included in SCING to further inform on regulatory structure to refine and improve on GRNs.[9,60–63] Information from multiple data types will become an integral part of the systems biology, and future efforts to properly model multiomics data simultaneously to inform on complex disease are warranted.

Limitations of the study

Although we show high utility of SCING through a number of evaluation methods, the following limitations should be considered when applying the method. First, SCING infers GRNs based on observational scRNA-seq and spatial transcriptomics data and the links between genes are not necessarily causal or directional. Second, there are many hyperparameters that require selection, and while we show robustness across a range of hyperparameters, selection can be dataset dependent and optimization of hyperparameters is encouraged. Thirdly, currently the spatial information of spatial transcriptomics is not explicitly utilized in SCING GRN construction despite the fact that the resulting GRNs show spatial patterns. We finally note that additional omics information (i.e., scATAC-seq, scHi-C, ChIP-seq, or cis/*trans* eQTL, etc.) can provide additional regulatory information that would improve GRN accuracy and will be considered in future development of SCING.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - Lead contact
  - Materials availability
  - Data and code availability
- METHOD DETAILS
  - SCING method overview
  - Preprocessing
  - Supercell construction
  - GRN inference in SCING
  - SCING parameter selection
  - SCING parameter benchmark framework and selection
  - Merging GRNs from data subsamples
- QUANTIFICATION AND STATISTICAL ANALYSIS
  - Other GRN methods
  - Datasets

- Computation of time requirements
- Overview of network robustness evaluation based on perturb-seq datasets
- Computation of guide RNA perturbation coefficients
- Determination of significant perturbation effects
- Selection of genes and cells for perturb-seq networks
- Evaluation of perturbation predictions
- Network robustness evaluation based on training and held out testing data
- Computation of network characteristics
- Computation of network overlap
- Assessment of disease subnetwork retrieval of GRNs
- Application of SCING to construct GRNs for all MCA cell types and assessment of network relevance to all DisGeNET disease gene sets
- Biological application to microglia in Alzheimer's disease patients
- Batch correction comparison

- APPLICATION OF SCING TO VISIUM SPATIAL TRANSCRIPTOMICS DATA FOR MOUSE AD AND WT BRAIN

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.isci.2023.107124.

ACKNOWLEDGMENTS

This work is supported by NIH/NINDS R01s NS117148 and NS111378, NIH T32s LM012424 and HL139450, and the UCLA Dissertation Year Fellowship.

AUTHOR CONTRIBUTIONS

R.L. and X.Y designed the SCING algorithm and evaluation framework. R.L. implemented the SCING method, evaluation framework, and benchmarked the approaches. M.C. performed hyperparameter benchmarking for SCING. N.W. built network related figures in Cytoscape. C.P. advised on Alzheimer's and neuroscience related results. R.L., M.C., N.W., and X.Y. wrote and edited the manuscript.

DECLARATION OF INTERESTS

We have nothing to declare.

Received: October 17, 2022
Revised: March 31, 2023
Accepted: June 9, 2023
Published: June 14, 2023

REFERENCES

1. Somvanshi, P.R., and Venkatesh, K.V. (2014). A conceptual review on systems biology in health and diseases: from biological networks to modern therapeutics. Syst. Synth. Biol. *8*, 99–116.

2. Yan, J., Risacher, S.L., Shen, L., and Saykin, A.J. (2018). Network approaches to systems biology analysis of complex disease: integrative methods for multi-omics data. Brief. Bioinform. *19*, 1370–1381.

3. Hood, L., and Tian, Q. (2012). Systems approaches to biology and disease enable translational systems medicine. Dev. Reprod. Biol. *10*, 181–185.

4. Yang, X. (2020). Multitissue Multiomics Systems Biology to Dissect Complex Diseases. Trends Mol. Med. *26*, 718–728.

5. Chen, J.C., Alvarez, M.J., Talos, F., Dhruv, H., Rieckhof, G.E., Iyer, A., Diefes, K.L., Aldape, K., Berens, M., Shen, M.M., and Califano, A. (2014). Identification of causal genetic drivers of human disease through systems-level analysis of regulatory networks. Cell *159*, 402–414.

6. Chatterjee, S., and Ahituv, N. (2017). Gene Regulatory Elements, Major Drivers of Human Disease. Annu. Rev. Genom. Hum. Genet. *18*, 45–63.

7. Zhu, J., Zhang, B., Smith, E.N., Drees, B., Brem, R.B., Kruglyak, L., Bumgarner, R.E., and Schadt, E.E. (2008). Integrating large-scale functional genomic data to dissect the complexity of yeast regulatory networks. Nat. Genet. *40*, 854–861. https://doi.org/10.1038/ng.167.

8. Schadt, E.E., Lamb, J., Yang, X., Zhu, J., Edwards, S., Guhathakurta, D., Sieberts, S.K., Monks, S., Reitman, M., Zhang, C., et al. (2005). An integrative genomics approach to infer causal associations between gene expression and disease. Nat. Genet. *37*, 710–717.

9. Zhu, J., Lum, P.Y., Lamb, J., GuhaThakurta, D., Edwards, S.W., Thieringer, R., Berger, J.P., Wu, M.S., Thompson, J., Sachs, A.B., and Schadt, E.E. (2004). An integrative genomics approach to the reconstruction of gene networks in segregating populations. Cytogenet. Genome Res. *105*, 363–374.

10. Nomura, S. (2021). Single-cell genomics to understand disease pathogenesis. J. Hum. Genet. *66*, 75–84.

11. Haque, A., Engel, J., Teichmann, S.A., and Lönnberg, T. (2017). A practical guide to single-cell RNA-sequencing for biomedical

research and clinical applications. Genome Med. *9*, 75.

12. Rao, A., Barkley, D., França, G.S., and Yanai, I. (2021). Exploring tissue architecture using spatial transcriptomics. Nature *596*, 211–220.

13. Pratapa, A., Jalihal, A.P., Law, J.N., Bharadwaj, A., and Murali, T.M. (2020). Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. Nat. Methods *17*, 147–154.

14. Blencowe, M., Arneson, D., Ding, J., Chen, Y.W., Saleem, Z., and Yang, X. (2019). Network modeling of single-cell omics data: challenges, opportunities, and progresses. Emerg. Top. Life Sci. *3*, 379–398.

15. Kang, Y., Thieffry, D., and Cantini, L. (2021). Evaluating the Reproducibility of Single-Cell Gene Regulatory Network Inference Algorithms. Front. Genet. *12*, 617282.

16. Kim, S. (2015). An R Package for a Fast Calculation to Semi-partial Correlation Coefficients. Commun. Stat. Appl. Methods *22*, 665–674.

17. Chan, T.E., Stumpf, M.P.H., and Babtie, A.C. (2017). Gene Regulatory Network Inference from Single-Cell Data Using Multivariate Information Measures. Cell Syst. *5*, 251–267.e3.

18. Moerman, T., Aibar Santos, S., Bravo González-Blas, C., Simm, J., Moreau, Y., Aerts, J., and Aerts, S. (2019). GRNBoost2 and Arboreto: efficient and scalable inference of gene regulatory networks. Bioinformatics *35*, 2159–2161.

19. Aibar, S., González-Blas, C.B., Moerman, T., Huynh-Thu, V.A., Imrichova, H., Hulselmans, G., Rambow, F., Marine, J.C., Geurts, P., Aerts, J., et al. (2017). SCENIC: single-cell regulatory network inference and clustering. Nat. Methods *14*, 1083–1086.

20. Schvartzman, J.M., Thompson, C.B., and Finley, L.W.S. (2018). Metabolic regulation of chromatin modifications and gene expression. J. Cell Biol. *217*, 2247–2259.

21. Bannister, A.J., and Kouzarides, T. (2011). Regulation of chromatin by histone modifications. Cell Res. *21*, 381–395.

22. Martin, E.W., and Sung, M.-H. (2018). Challenges of Decoding Transcription Factor Dynamics in Terms of Gene Regulation. Cells *7*, 132. https://doi.org/10.3390/cells7090132.

23. Matsumoto, H., Kiryu, H., Furusawa, C., Ko, M.S.H., Ko, S.B.H., Gouda, N., Hayashi, T., and Nikaido, I. (2017). SCODE: an efficient regulatory network inference algorithm from single-cell RNA-Seq during differentiation. Bioinformatics *33*, 2314–2321.

24. Qiu, X., Rahimzamani, A., Wang, L., Ren, B., Mao, Q., Durham, T., McFaline-Figueroa, J.L., Saunders, L., Trapnell, C., and Kannan, S. (2020). Inferring Causal Gene Regulatory Networks from Coupled Single-Cell Expression Dynamics Using Scribe. Cell Syst. *10*, 265–274.e11.

25. Deshpande, A., Chu, L.-F., Stewart, R., and Gitter, A. (2022). Network inference with Granger causality ensembles on single-cell transcriptomics. Cell Rep. *38*, 110333.

26. Han, X., Wang, R., Zhou, Y., Fei, L., Sun, H., Lai, S., Saadatpour, A., Zhou, Z., Chen, H., Ye, F., et al. (2018). Mapping the Mouse Cell Atlas by Microwell-Seq. Cell *173*, 1307.

27. Morabito, S., Miyoshi, E., Michael, N., Shahin, S., Martini, A.C., Head, E., Silva, J., Leavy, K., Perez-Rosendahl, M., and Swarup, V. (2021). Single-nucleus chromatin accessibility and transcriptomic characterization of Alzheimer's disease. Nat. Genet. *53*, 1143–1155.

28. Chen, W.-T., Lu, A., Craessaerts, K., Pavie, B., Sala Frigerio, C., Corthout, N., Qian, X., Laláková, J., Kühnemund, M., Voytyuk, I., et al. (2020). Spatial Transcriptomics and In Situ Sequencing to Study Alzheimer's Disease. Cell *182*, 976–991.e19. https://doi.org/10.1016/j.cell.2020.06.038.

29. Chen, G., Ning, B., and Shi, T. (2019). Single-Cell RNA-Seq Technologies and Related Computational Data Analysis. Front. Genet. *10*, 317.

30. Hwang, B., Lee, J.H., and Bang, D. (2018). Single-cell RNA sequencing technologies and bioinformatics pipelines. Exp. Mol. Med. *50*, 1–14.

31. Papalexi, E., Mimitou, E.P., Butler, A.W., Foster, S., Bracken, B., Mauck, W.M., 3rd, Wessels, H.H., Hao, Y., Yeung, B.Z., Smibert, P., and Satija, R. (2021). Characterizing the molecular regulation of inhibitory immune checkpoints with multimodal single-cell screens. Nat. Genet. *53*, 322–331.

32. Dixit, A., Parnas, O., Li, B., Chen, J., Fulco, C.P., Jerby-Arnon, L., Marjanovic, N.D., Dionne, D., Burks, T., Raychowdhury, R., et al. (2016). Perturb-Seq: Dissecting Molecular Circuits with Scalable Single-Cell RNA Profiling of Pooled Genetic Screens. Cell *167*, 1853–1866.e17.

33. Langfelder, P., and Horvath, S. (2008). WGCNA: an R package for weighted correlation network analysis. BMC Bioinf. *9*, 559.

34. Golbeck, J. (2015). Introduction to Social Media Investigation: A Hands-On Approach (Syngress).

35. Huang, J.K., Carlin, D.E., Yu, M.K., Zhang, W., Kreisberg, J.F., Tamayo, P., and Ideker, T. (2018). Systematic Evaluation of Molecular Networks for Discovery of Disease Genes. Cell Syst. *6*, 484–495.e5.

36. Piñero, J., Saüch, J., Sanz, F., and Furlong, L.I. (2021). The DisGeNET cytoscape app: Exploring and visualizing disease genomics data. Comput. Struct. Biotechnol. J. *19*, 2960–2967.

37. Janeway, C.A., Jr., Travers, P., Walport, M., and Shlomchik, M.J. (2001). Principles of Innate and Adaptive Immunity (Garland Science).

38. Keren-Shaul, H., Spinrad, A., Weiner, A., Matcovitch-Natan, O., Dvir-Szternfeld, R., Ulland, T.K., David, E., Baruch, K., Lara-Astaiso, D., Toth, B., and Itzkovitz, S. (2017). A unique microglia type associated with restricting development of alzheimer's disease. Cell *169*, 1276–1290.e17. https://doi.org/10.1016/j.cell.2017.05.018.

39. Hemonnot, A.-L., Hua, J., Ulmann, L., and Hirbec, H. (2019). Microglia in Alzheimer Disease: Well-Known Targets and New Opportunities. Front. Aging Neurosci. *11*, 233.

40. Traag, V.A., Waltman, L., and van Eck, N.J. (2019). From Louvain to Leiden: guaranteeing well-connected communities. Sci. Rep. *9*, 5233.

41. Haghverdi, L., Lun, A.T.L., Morgan, M.D., and Marioni, J.C. (2018). Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. Nat. Biotechnol. *36*, 421–427.

42. Korsunsky, I., Millard, N., Fan, J., Slowikowski, K., Zhang, F., Wei, K., Baglaenko, Y., Brenner, M., Loh, P.R., and Raychaudhuri, S. (2019). Fast, sensitive and accurate integration of single-cell data with Harmony. Nat. Methods *16*, 1289–1296.

43. Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck, W.M., 3rd, Hao, Y., Stoeckius, M., Smibert, P., and Satija, R. (2019). Comprehensive Integration of Single-Cell Data. Cell *177*, 1888–e1902.e21.

44. Tran, H.T.N., Ang, K.S., Chevrier, M., Zhang, X., Lee, N.Y.S., Goh, M., and Chen, J. (2020). A benchmark of batch-effect correction methods for single-cell RNA sequencing data. Genome Biol. *21*, 12.

45. Shin, J.-W., Kwon, M.A., Hwang, J., Lee, S.J., Lee, J.H., Kim, H.J., Lee, K.B., Lee, S.J., Jeong, E.M., Chung, J.H., and Kim, I.G. (2020). Roles of microglial membranes in Alzheimer's disease. Cell Death Dis. *11*, 301–314.

46. Bertram, L., and Tanzi, R.E. (2009). Genome-wide association studies in Alzheimer's disease. Hum. Mol. Genet. *18*, R137–R145.

47. Hooli, B.V., Mohapatra, G., Mattheisen, M., Parrado, A.R., Roehr, J.T., Shen, Y., Gusella, J.F., Moir, R., Saunders, A.J., Lange, C., and Tanzi, R.E. (2012). Role of common and rare APP DNA sequence variants in Alzheimer disease. Neurology *78*, 1250–1257.

48. Sherva, R., Tripodis, Y., Bennett, D.A., Chibnik, L.B., Crane, P.K., de Jager, P.L., Farrer, L.A., Saykin, A.J., Shulman, J.M., Naj, A., and Green, R.C. (2014). Genome-wide association study of the rate of cognitive decline in Alzheimer's disease. Alzheimers Dement. *10*, 45–52.

49. Bradl, M., and Lassmann, H. (2010). Oligodendrocytes: biology and pathology. Acta Neuropathol. *119*, 37–53.

50. Haddad-Tóvolli, R., Dragano, N.R.V., Ramalho, A.F.S., and Velloso, L.A. (2017). Development and Function of the

Blood-Brain Barrier in the Context of Metabolic Control. Front. Neurosci. *11*, 224.

51. Daigle, T.L., Madisen, L., Hage, T.A., Valley, M.T., Knoblich, U., Larsen, R.S., Takeno, M.M., Huang, L., Gu, H., Larsen, R., and Mills, M. (2018). A Suite of Transgenic Driver and Reporter Mouse Lines with Enhanced Brain-Cell-Type Targeting and Functionality. Cell *174*, 465–480.e22.

52. Vercruysse, P., Vieau, D., Blum, D., Petersén, Å., and Dupuis, L. (2018). Hypothalamic Alterations in Neurodegenerative Diseases and Their Relation to Abnormal Energy Metabolism. Front. Mol. Neurosci. *11*, 2.

53. Shen, Y., Tian, M., Zheng, Y., Gong, F., Fu, A.K.Y., and Ip, N.Y. (2016). Stimulation of the Hippocampal POMC/MC4R Circuit Alleviates Synaptic Plasticity Impairment in an Alzheimer's Disease Model. Cell Rep. *17*, 1819–1831.

54. Jais, A., Paeger, L., Sotelo-Hitschfeld, T., Bremser, S., Prinzensteiner, M., Klemm, P., Mykytiuk, V., Widdershooven, P.J., Vesting, A.J., Grzelka, K., and Minère, M. (2020). PNOCARC Neurons Promote Hyperphagia and Obesity upon High-Fat-Diet Feeding. Neuron *106*, 1009–1025.e10.

55. Jackson, C.A., Castro, D.M., Saldi, G.-A., Bonneau, R., and Gresham, D. (2020). Gene regulatory network reconstruction using single-cell RNA sequencing of barcoded genotypes in diverse environments. Elife *9*, e51254.

56. Zhang, B., and Zhu, J. (2013). Identification of key causal regulators in gene networks. In Proceedings of the World Congress on Engineering, *2*Proceedings of the World Congress on Engineering, pp. 5–8.

57. Nussinov, R., Jang, H., Tsai, C.-J., and Cheng, F. (2019). Review: Precision medicine and driver mutations: Computational methods, functional assays and conformational principles for interpreting cancer drivers. PLoS Comput. Biol. *15*, e1006658. https://doi.org/10.1371/journal.pcbi.1006658.

58. Dugger, S.A., Platt, A., and Goldstein, D.B. (2018). Drug development in the era of precision medicine. Nat. Rev. Drug Discov. *17*, 183–196.

59. Emmert-Streib, F., Dehmer, M., and Haibe-Kains, B. (2014). Gene regulatory networks and their applications: understanding biological and medical problems in terms of networks. Front. Cell Dev. Biol. *2*, 38.

60. Jansen, C., Ramirez, R.N., El-Ali, N.C., Gomez-Cabrero, D., Tegner, J., Merkenschlager, M., Conesa, A., and Mortazavi, A. (2019). Building gene regulatory networks from scATAC-seq and scRNA-seq using Linked Self Organizing Maps. PLoS Comput. Biol. *15*, e1006555.

61. Wang, C., Sun, D., Huang, X., Wan, C., Li, Z., Han, Y., Qin, Q., Fan, J., Qiu, X., Xie, Y., and Meyer, C.A. (2020). Integrative analyses of single-cell transcriptome and regulome using MAESTRO. Genome Biol. *21*, 198.

62. Zhang, R., Zhou, T., and Ma, J. (2022). Multiscale and integrative single-cell Hi-C analysis with Higashi. Nat. Biotechnol. *40*, 254–261.

63. Albert, F.W., Bloom, J.S., Siegel, J., Day, L., and Kruglyak, L. (2018). Genetics of trans-regulatory variation in gene expression. Elife *7*, e35471.

64. Van Rossum, G., and Drake, F.L. (2009). Python 3 Reference Manual: (Python Documentation Manual Part 2) (CreateSpace Independent Publishing Platform).

65. Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N., et al. (2020). Array programming with NumPy. Nature *585*, 357–362.

66. McKinney, W. (2010). Data Structures for Statistical Computing in Python. In Proceedings of the 9th Python in Science Conference (SciPy). https://doi.org/10.25080/majora-92bf1922-00a.

67. Hunter, J.D. (2007). A 2D Graphics Environment. Comput. Sci. Eng. *9*, 90–95.

68. Wolf, F.A., Angerer, P., and Theis, F.J. (2018). large-scale single-cell gene expression data analysis. Genome Biol. *19*, 15.

69. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Muller, A., Nothman, J., Louppe, G., et al. (2012). Scikit-learn: Machine Learning in Python. Prepint at. arXiv, 2825–2830.

70. Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, C., Burvoski, E., Peterson, P., Weckesser, W., Bright, J., et al. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat. Methods *17*, 261–272.

71. Seabold, S., and Perktold, J.S. (2010). Econometric and statistical modeling with python. In Proceedings of the 9th Python in Science Conference (SciPy). https://doi.org/10.25080/majora-92bf1922-011.

72. Rocklin, M.D. (2015). Parallel Computation with Blocked algorithms and Task Scheduling. In Proceedings of the 14th Python in Science Conference (SciPy). https://doi.org/10.25080/majora-7b98e3ed-013.

73. Foster, P. Pyitlib: A Library of Information-Theoretic Methods for Data Analysis and Machine Learning, Implemented in Python and NumPy. (Github)

74. Csárdi, G. & Nepusz, T. The Igraph Software Package for Complex Network Research. (2006).

75. Hagberg, A.a., Schult, D.a., and Swart, P.j. (2008). Exploring network structure, dynamics, and function using NetworkX. In Proceedings of 7th Python in Science Conference (SciPy2008), G. Varoquaux, T. Vaught, and J. Millman, eds. (scientific research publishing), pp. 11–15. https://www.scirp.org/(S(351jmbntvnsjt1aadkposzje))/reference/ReferencesPapers.aspx?ReferenceID=405649.

76. Bravo González-Blas, C., Minnoye, L., Papasokrati, D., Aibar, S., Hulselmans, G., Christiaens, V., Davie, K., Wouters, J., and Aerts, S. (2019). cisTopic: cis-regulatory topic modeling on single-cell ATAC-seq data. Nat. Methods *16*, 397–400.

77. R Core Team (2021). R A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna (scientific research publishing). https://www.scirp.org/(S(czeh2tfqw2orz553k1w0r45))/reference/referencespapers.aspx?referenceid=3131254.

78. Kuleshov, M.V., Jones, M.R., Rouillard, A.D., Fernandez, N.F., Duan, Q., Wang, Z., Koplev, S., Jenkins, S.L., Jagodnik, K.M., Lachmann, A., et al. (2016). Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. Nucleic Acids Res *44*, W90–W97.

79. Piñero, J., Ramírez-Anguita, J.M., Saüch-Pitarch, J., Ronzano, F., Centeno, E., Sanz, F., and Furlong, L.I. (2020). The DisGeNET knowledge platform for disease genomics: 2019 update. Nucleic Acids Res. *48*, D845–D855.

80. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Bertrand, T., Grisel, O., Blondel, M., Peter, P., Weiss, R., Vincent, D., et al. (2011). Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. *12*, 2825–2830.

81. Adler, A.I., and Painsky, A. (2022). Feature Importance in Gradient Boosting Trees with Cross-Validation Feature Selection. Entropy *24*, 687.

82. Kubkowski, M., Mielniczuk, J., and Teisseyre, P. (2021). How to gain on power: novel conditional independence tests based on short expansion of conditional mutual information. J. Mach. Learn. Res. *22*, 2877–2933.

83. Satija, R., Farrell, J.A., Gennert, D., Schier, A.F., and Regev, A. (2015). Spatial reconstruction of single-cell gene expression data. Nat. Biotechnol. *33*, 495–502.

84. Gustavsen, J.A., Pai, S., Isserlin, R., Demchak, B., and Pico, A.R. (2019). RCy3: Network biology using Cytoscape from within R. F1000Res. *8*, 1774.

85. Yuste, R., Hawrylycz, M., Aalling, N., Aguilar-Valles, A., Arendt, D., Armananzas, R., Ascoli, G.A., Bielza, C., Bokharaie, V., Borgtoft Bergmann, T., et al. (2020). A community-based transcriptomics classification and nomenclature of neocortical cell types. Nat. Neurosci. *23*, 1456–1468.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Software and algorithms		
Python 3.8.13	Van Rossum et al.[64]	https://www.python.org/
Numpy 1.21.6	Harris et al.[65]	https://numpy.org/
Pandas 1.4.2	McKinney et al.[66]	https://pandas.pydata.org/
Matplotlib 3.5.2	Hunter et al.[67]	https://matplotlib.org/
Scanpy 1.9.1	Wolf et al.[68]	https://github.com/scverse/scanpy
Scikit-learn 1.1.1	Pedregosa et al.[69]	https://scikit-learn.org/stable/index.html
Scipy 1.8.1	Virtanen et al.[70]	https://scipy.org/
Statsmodels 0.13.2	Seabold et al.[71]	https://www.statsmodels.org/stable/index.html
Dask/Distributed 2022.7.0	Rocklin et al.[72]	https://dask.org
Pyitlib 0.2.2	Peter Foster and Michael Milton[73]	https://github.com/pafoster/pyitlib
Leidenalg 0.8.10	Traag et al.[40]	https://leidenalg.readthedocs.io/en/stable/
Python_igraph 0.9.11	Csardi et al.[74]	https://igraph.org/
Networkx 2.8.3	Hagberg et al.[75]	https://networkx.org/
Ctxcore 0.1.1	Bravo González-Blas et al.[76]	https://ctxcore.readthedocs.io/en/latest/
Pyscenic 0.11.2	Aibar et al.[19]	https://scenic.aertslab.org/
R 4.1.2	R Core Team (2022).[77]	https://www.r-project.org/
enrichR 3.0	Kuleshov et al.[78]	https://www.rdocumentation.org/packages/enrichR/versions/3.0
Ppcor	Kim, 2015[16]	https://www.rdocumentation.org/packages/ppcor/versions/1.1
PIDC	Chan et al., 2017[17]	https://github.com/Tchanders/NetworkInference.jl
GRNBOOST2	Moerman et al., 2019[18]	https://arboreto.readthedocs.io/en/latest/installation.html
Random Walk	Huang et al., 2018[35]	https://github.com/idekerlab/Network_Evaluation_Tools
SCING	This paper/GitHub	https://github.com/XiaYangLabOrg/SCING
Deposited data		
Human Alzheimer's Disease snRNAseq	Morabito et al., 2021[27]	https://www.synapse.org/#!Synapse:syn22079621/
Dendritic cell and k562 cell line perturb-seq datasets	Dixit et al., 2016[32]	GEO: GSE90063
THP-1 cells	Papalexi et al., 2021[31]	GEO: GSE153056
Mouse Cell Atlas	Han et al., 2018[26]	https://bis.zju.edu.cn/MCA/
DisGeNET	Piñero et al., 2019[79]	http://www.disgenet.org/
Mouse AD Visium	Chen et al., 2020[28]	GEO: GSE152506

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources should be directed to and will be fulfilled by the lead contact, Xia Yang (xyang123@g.ucla.edu).

Materials availability

This study did not generate new reagents.

Data and code availability

- All data and code generated in this study can be found at: https://github.com/XiaYangLabOrg/SCING, or through contact of the lead author, Xia Yang (xyang123@g.ucla.edu).
- Other publicly available data used can be found through downloading from their respective repositories.

METHOD DETAILS

SCING method overview

To reduce the challenges from data sparsity from single cell omics, as well as reduce computational time, we first used supercells which combine gene expression data from subsets of cells sharing similar transcriptome patterns. To improve the robustness of GRNs, we built GRNs on subsamples before merging the networks, keeping edges that appear in at least 20% of networks. Lastly, we removed cycles and bidirectional edges where one direction was >25% stronger than the other direction and pruned the network using conditional mutual information to reduce redundancy. These steps are described in more detail below. To benchmark the performance of SCING, we selected three existing methods, namely GRNBOOST2, ppcor, and PIDCm with default parameters. These methods were selected based on previous benchmarking studies where superior performance of these methods were supported.[13]

Preprocessing

For supercell preprocessing, we first normalized the data for a total count number of 10,000 per cell using the pp.normalize_total function from Scanpy.[68] We then took the log1p of the gene expression values, using Scanpy's pp.log1p function. We then identified and subset to the top 2000 highly variable genes using Scanpy's pp.highly_variable_genes function.[68] Data were then centered and scaled using Scanpy's pp.scale function and further used to compute the nearest neighbor embedding with the default 10 neighbors using Scanpy's pp.neighbors function on the top 20 principal components of the gene expression, calculated using Scanpy's tl.pca function.[68]

For network building preprocessing, we normalized the total number of counts in each cell to 10,000 using Scanpy's pp.normalize_total function and took the natural log of the gene expression using Scanpy's pp.log1p function.[68] We removed genes not expressed in any cells and any duplicate genes. We transposed, centered, and scaled the data using Scanpy's pp.scale function before running principal component analysis (PCA) using Scanpy's tl.pca function.[68] This provides us with low dimensional embeddings for each gene. A nearest neighbor algorithm from scikit-learn[80] was used to find the nearest neighbors of each gene. The potential regulatory relationship between genes was limited to the 100 nearest neighbors for each downstream gene.

Supercell construction

We define a supercell as a pseudobulk generated from a cluster of cells, determined through Leiden clustering on the nearest neighbors connectivities graph of cells in the principal component space of the gene expression data. For each dataset, we perform preprocessing as described above. We then used the Leiden graph partitioning algorithm to separate cells into groups using Scanpy's tl.leiden function.[68] The Leiden resolution is determined by the user input specifying the desired final number of supercells using binary search on the resolution parameter, with higher resolution leading to more supercells and lower resolution leading to fewer supercells. Here, we used 500 supercells, which balances runtime, with dataset summarization (Figure S14). Benchmarks for run time, supercell mean and variance in sparsity, and network robustness suggested 500 supercells to be optimal for balancing the tradeoff between sparsity mitigation and network performance (Figure S14). We then merged each group of cells into a supercell by averaging the gene expression within each group.

GRN inference in SCING

For each downstream gene, we trained a gradient boosting regressor[18] to predict the expression using nearest neighbors as potential upstream regulators. For each upstream gene, we form a directed edge to the downstream gene with a corresponding edge weight based on the feature importance of that

upstream gene in predicting the downstream gene in the gradient boosting regressor. We keep the top 10 percent of edges based on edge weight to remove edges with very low weights. For each gradient boosting regressor, we used 500 estimators, max depth of 3, learning rate of 0.01, 90 percent subsample, the square root of the total number of features as max features, and an early stop window of 25 trees, if the regressor was no longer showing improved performance.[18]

SCING parameter selection

Parameter selection was used to retain biological accuracy while limiting computational cost. We determined that using 500 supercells and 100 GRNs per merged network balanced computational resourcefulness with robustness. If computational cost is not an issue, supercells are not necessary and more GRNs can be built as intermediates. Since GRNs are typically built within a single cell type, we use 10 principal components (PCs) for determining gene covariance. Typically in scRNAseq analysis more PCs are used, but there is less variation overall within one cell type. We determined 100 neighbors to be chosen for each gene, again balancing computational cost.

SCING parameter benchmark framework and selection

SCING contains four tunable hyperparameters: the number of supercells, the number of subsampled networks for bootstrapping, the number of nearest neighbors for feature selection in gradient boosting regression, and the consensus edge overlap threshold when merging subsampled networks. We designed a pipeline to compare computational efficiency, network properties, and GRN robustness in gene expression prediction accuracy across different parameter settings (Figure 1C). Optimal parameters were defined based on the balance of these metrics, ideally resulting in fast run time, more genes in the network, and high prediction accuracy. We tested one parameter at a time while fixing others at their default values. We tested the number of supercells at 100, 300, 500, 700, and 900 (Figure S14); number of nearest neighbors at 30, 50, 100, 200, and 400 (Figure S15); number of subsampled networks at 30, 50, 100, 200, 300, and 400 (Figure S16); and edge consensus threshold proportions at 0.05, 0.1, 0.2, 0.5, and 0.8 (Figure S17). Networks were generated under these settings on all genes in the oligodendrocytes from Morabito et al.27, since this cell type contained the most cells in the dataset, for building networks that are representative of SCING performance. We performed network robustness on prediction accuracy with the same protocol used to compare the cosine similarities of predicted and actual gene expression between SCING and other GRN methods, which is detailed in "Network robustness evaluation based on training and held out testing data" in the methods section. The network characteristics and run times were calculated under the same framework as the cross-method comparisons, which is detailed in "Computation of time requirements" in the methods section.

As the supercells were used to mitigate gene sparsity, we used low mean supercell gene sparsity and low variance in supercell gene sparsity as an initial guide for our selection of 500 supercells. The benchmark results coincided with this general heuristic, since 500 supercells exhibited a balance of high relative cosine similarity for target gene expression prediction and reasonable GRN construction run time (Figure S14). Given our robust supercell selection, the number of subsampled networks had little effect on SCING performance, with any subsample size above 30 networks yielding consistent accuracy. Run times generally increased with subsampled networks, but the networks were largest at 100 subsampled networks. While any size above 30 is an acceptable choice, we decided on 100 subsampled networks to balance run time and network size (Figure S16). For the nearest neighbor selection, we observed comparable cosine similarity across all parameter values, faster run time for GRNs with lower numbers of neighbors, and denser networks with higher numbers of neighbors (Figure S15). To balance efficiency and network size, we decided the default setting to be 100 neighbors. Lastly, our consensus parameter benchmark revealed that GRNs with higher edge consensus thresholds yield smaller and sparser networks with lower gene counts, quicker run times, and slightly improved gene prediction accuracy and overfitting. While stringent thresholds remove edges that would reduce overfitting and increase prediction accuracy for downstream genes in the networks, they run the risk of losing regulatory information, since these smaller networks omit many genes and cannot predict their expression. In order to comprehensively explain a cell type's regulatory landscape, we decided to include more genes in our networks without greatly sacrificing performance (Figure S17). For these reasons, we set the consensus threshold to 20% for our study.

Merging GRNs from data subsamples

We subsampled supercells from the supercell data without replacement and built one network for each subsample. For each subsampled network, we kept the top 10 percent of edges based on the edge weights, which are the feature importance measures of upstream genes in predicting downstream genes in the trained gradient boosting regressor, to reduce the number of edges with very low edge weights. We also kept the top 3 edges for each downstream gene to reduce any gene specific bias caused by feature importance.[81] Of these edges, we kept the edges that appeared in at least 20% of all networks from the subsamples into a merged network. The threshold of 20% was based on the testing results of multiple thresholds (Figure S17), which showed this threshold balances accuracy, network size, and run time. For the edges that were kept in the merged network, the edge weights were the sum of the weights for that edge across the subsampled networks. We also removed reversed edges if the edge with a higher weight was at least 25% stronger than that of the weaker reverse direction. Otherwise, we kept the edge bidirectional. We removed cycles with more than two edges in the graph by removing the edge with the lowest edge weight. Additionally, we removed triads in the network based on the significance of the conditional mutual information to remove redundant edges between genes that may result primarily from sharing an upstream regulator in the network. The p value of the conditional mutual information was based on the chi-squared distribution.[82] If an edge between two genes was not statistically significant given a parent of both of the genes, then the edge was removed.

QUANTIFICATION AND STATISTICAL ANALYSIS

Other GRN methods

We benchmarked SCING against GRNBOOST2, ppcor, and PIDC. Default parameters were used for all existing approaches unless otherwise specified.

For GRNBOOST2,[18] we ran this approach by predicting the expression of all genes from all other genes. We then took the top 10% of edges to reduce the number of edges with extremely low importance (e.g. 10^{-17}).

For ppcor,[16] due to the sparse non-linear nature of scRNAseq connections, we ran the approach using spearman correlation. We only kept edges with a Benjamini-Hochberg FDR <0.05.

For PIDC,[17] according to their tutorial, we used a threshold of 0.1, to keep the top 10% of highest scoring edges.

Datasets

For the Perturb-seq validation, we used datasets from Dixit et al.[32] and Papalexi et al.[31] Dixit et al. has 24 transcription factors perturbed in dendritic cells and 25 cell cycle genes targeted in K562 cells, while Papalexi et al. has 25 PD-L1 regulators perturbed in THP-1 cells.

For the train-test split and network consistency assessment, we used the human AD snRNAseq data from Morabito et al.[27] This adds another slightly different data type from the scRNAseq in the perturb-seq and MCA and is later used for biological application with microglial cells in AD.

We used the mouse cell atlas[26] scRNAseq database, since it has a large number of cell types (106 cell types) across numerous tissues (33 tissues), to test 446 disease associations with the random walk approach from Huang et al.[35] This additionally provides a resource of GRNs throughout cell types of the entire mouse.

Finally, to test the applicability of SCING on spatial transcriptomics data, we used the mouse AD dataset from Chen et al.[28] This dataset contains AD and WT mice from various age groups (3, 6, 12, 18 months), in addition to amyloid beta plaque staining.

Computation of time requirements

We determined the run time to build a GRN from each approach on subsets of cells and genes with varying cell numbers and gene numbers. All tests were performed on a ryzen 9 3900X 12-Core processor with 64Gb RAM. We determined the speed on 10 iterations of randomly selected genes (1000, 2000, 4000) with 1000 cells each, and on randomly selected cells (250, 500, 1000) with 1000 genes each.

Overview of network robustness evaluation based on perturb-seq datasets

Briefly, to determine the accuracy of the GRNs from each approach with Perturb-seq data, we first identified significantly altered genes downstream of each guide RNA perturbation through an elastic net regression approach,[32] as detailed below. We then determine the accuracy of a given network by identifying the true positive rate (TPR) and false positive rate (FPR) at each depth in the network. The AUROC and TPR at FPR 0.05 were determined for each network on a given perturbation. More details on each step are below.

Computation of guide RNA perturbation coefficients

First, we downloaded the Perturb-seq data from Papalexi et al.[31] and Dixit et al.[32] We then followed the steps as described by Dixit et al.[32] to compute guide RNA perturbation coefficients, which indicate the effects of a specific guide RNA (single perturbation) on other genes.

As cell state can affect gene perturbation efficiency, to determine cell states and remove state specific perturbations, we first separated out unperturbed cells which were not transfected with positive guide RNAs. We then clustered these unperturbed cells with Leiden clustering to define subclusters that represent cell states. The Leiden resolution was determined by identifying unique subclusters in the data (DC 0h: 1.2, DC 3h, 0.7, K562 cell cycle: 1.0, K562 TF: 1.0, Papalexi: 1.2). We then subset the data to highly variable genes (min_mean=0.0125, max_mean=3, min_disp=0.5), centered and scaled the highly variable genes data, and performed PCA to get the first 50 PCs. We then trained a linear support vector machine (SVM, C=1) on the top 50 PCs of the data to predict the cell subcluster (state) membership of the non-perturbed cells. We then applied the trained SVM to all cells that include both perturbed and non-perturbed cells to get a continuous probability of cell state for each cell. We used these continuous state probabilities in our regression equation to regress out state specific effects on gene expression. To identify the perturbation effect of a given guide RNA on genes other than the target gene, we utilized elastic net regression (l1_ratio=0.5, alpha=0.0005).[32] We generated a binary matrix (cells x RNA guides) which depicts which guide RNAs are in each cell based on the perturb-seq sequencing data. We then fit an elastic net model to predict the gene expression of all genes from the binary matrix in each cell, combined with the continuous state values determined for each cell. To remove the effect of synergistic perturbations, we removed cells with multiple perturbations. We determined each guide's perturbation effect on a given gene by the regression coefficient.

Determination of significant perturbation effects

To determine the significance of a given perturbation coefficient, we employed a permutation test as in Dixit et al.[32] For each guide, we permuted the vector of perturbations to randomize which cells received the given perturbation of interest. The elastic net regression model was trained with the same hyperparameters to determine the coefficients of perturbation. This approach was repeated 100 times to generate a null distribution of the perturbation effect of a given guide on each gene. The p-value was calculated as the fraction of null coefficients that were greater than or less than the true coefficient, determined by the sign of the coefficient. Significant perturbations were determined at a false discovery rate of 0.05 using the Benjamini-Hochberg procedure. This permutation approach was repeated for each guide RNA. A gene was determined as a downstream perturbation if at least one guide had a significant perturbation for the given gene.

Selection of genes and cells for perturb-seq networks

To reduce computational cost and enable network building for all approaches, we first took the top 3,000 highly variable genes using the variance stabilizing transform method,[83] including the differentially perturbed genes. We built two networks for each dataset, one using all cells and the other using only cells with non-zero expression of the gene of interest.

Evaluation of perturbation predictions

We built networks for each perturbed gene separately using SCING, GRNBOOST2, ppcor, and PIDC. Starting from the perturbed gene of interest, at each depth in the network, we determined the TPR and FPR based on the perturbed genes computed above. This gives a TPR vs FPR graph, from which an AUROC was computed. For each perturbed gene, we calculated the AUROC and the TPR at a FPR of 0.05.

Network robustness evaluation based on training and held out testing data

For each cell type in the Morabito et al dataset, we performed a train test split (50/50) of the cells and generated 10 sets of random subsamples of 3000 genes. We built a GRN on each gene subsample set in the training data and trained a gradient boosting regressor to predict the expression of each gene based on the gene expression of the predicted regulatory parents in the given GRN. We used the trained gradient boosting regressor to predict the expression of the 3,000 genes in the test dataset and evaluated the performance based on the cosine similarity metric between the actual gene expression and predicted gene expression. We generated the cosine similarity values on the training and testing data separately and computed the test to train ratio of the cosine similarity, with a smaller test to train ratio indicating potential overfitting of the training data.

Computation of network characteristics

We built GRNs on oligodendrocytes, astrocytes, and microglia from snRNAseq data from Morabito et al[27] For each cell type, we randomly selected 3000 genes (reduce computational time of methods) for each sample and generated 10 GRNs, in which 3000 genes were randomly subsampled from the full transcriptome. To compute scale-free network characteristics for each network, we fit a linear regression model on the log of each node degree with the log of the proportion of nodes at each degree. We removed low degree data points that are an artifact of scRNAseq sparsity. We also characterized each network by the number of edges in the network, number of genes remaining in the resulting network, and the mean betweenness centrality of nodes across the network.

Computation of network overlap

We used the Morabito et al. datasets described above and split each dataset in half and generated GRNs on each subset of cells. We checked the overlapping edges between the two networks and normalized for the expected number of overlapping edges based on the number of total edges in each network and the hypergeometric distribution. The overlap score measured the fraction of overlap between the two networks, divided by the expected number of overlapping edges.

Assessment of disease subnetwork retrieval of GRNs

We utilized a random walk approach from Huang et al.[35] to determine the ability for GRNs from different methods to accurately model disease gene subnetworks. This approach provides a biologically relevant benchmarking approach to determine a GRNs ability to model disease subnetworks. Briefly, the approach splits a known disease gene set into two groups, to attempt to reach the held out gene set starting from the selected disease genes through random walks. An improvement score is computed by calculating the z-score for a given network relative to 50 degree-preserved randomized networks.

We built networks from the MCA on immune cells from bone marrow, neurons from the brain, and hepatocytes from the liver. To accommodate less efficient tools, we subsetted the transcriptome to genes that are expressed in more than 5% of the cells in the dataset. We used the method from Huang et al. using relevant immune, neuronal, and metabolic disease gene sets from DisGeNET. We kept these genes with >5% percent expression and included the genes from the disease gene sets. We determined performance of each subnetwork based on the improved performance compared to the random network distribution.

Application of SCING to construct GRNs for all MCA cell types and assessment of network relevance to all DisGeNET disease gene sets

We applied SCING to all cell types for all tissues in the MCA and utilized the approach from Huang et al. to determine the ability of each network to accurately model each disease gene set. We clustered the disease gene sets and cell types using hierarchical clustering with complete linkage. We determined the number of disease sets accurately modeled by each cell type based on a performance gain of at least 0.1. We subsequently computed the number of cell types that can accurately model each disease set. To compare the number of diseases modeled by cell types from the adaptive and innate immune system on tissue relevant subsets of the DisGeNET diseases, we performed a t-test between the distributions of the number of disease gene sets each cell type can accurately model.

Biological application to microglia in Alzheimer's disease patients

We built a SCING network for the microglia on the genes expressed in at least 2.5 percent of cells in the Morabito et al. dataset.[27] For the SCING pipeline, we used 500 supercells, 70 percent of cells in each subsample, 100 neighbors, 10 PCs, and 100 subsamples. We utilized the Leiden graph partitioning algorithm to divide genes in the resulting GRNs into modules. We performed Leiden clustering at different resolutions and performed pathway enrichment analysis on the modules using the enrichr[78] R package, using the GO biological process, DisGeNET, Reactome, BioCarta, and KEGG knowledge bases. We selected the resolution (0.0011) that had the highest fraction of modules annotated for between 20 and 50 modules per network. This avoids clustering too many modules with few genes while maintaining enough separate modules to have biological interpretation. We used the AUCell method from the SCENIC workflow,[19] to retrieve module specific expressions (AUCell scores) for each cell. We found trait (diagnosis, plaque stage, tangle stage) associated modules by fitting a linear regression model to predict the trait based on the module score, while regressing out the effects of sex. For each trait, multiple testing was controlled at FDR <0.05 with the Benjamini-Hochberg procedure. The subnetwork for vesicle-mediated transport in module 2 was visualized using Cytoscape.[84] We determined marker genes using the Allen Brain Atlas whole brain Smartseq2 data.[85]

Batch correction comparison

We compared top batch correction methods from Seurat, Harmony, and fastMNN with SCING module embeddings. To evaluate each method, we determined the average proportion of cells with the same group assignment (sample, batch, diagnosis, tangle stage, plaque stage, and sex), using 20 PCs and a variable number of neighbors (0.25, 0.5, 1, 2, 4, 8, and 16 percent of the dataset) (Equation 1). We determined the ability of each approach to remove batch and sample specific differences while retaining biologically relevant differences (diagnosis, tangle stage, plaque stage, and sex) by removing the batch and sample differences with an F1-score[44] (Equation 2).

$$\text{neigh\_score} = \frac{1}{n_{cell}} \sum_{cells} \frac{similar\_neighbors}{n\_neighbors)} \quad \text{Equation 1}$$

neigh_score: neighborhood score used to find the average fraction of neighbors of the same type (i.e. batch).

similar_neighbors: number of neighbors of a given cell that have the same identity (i.e. batch)

n_neighbors: number of total neighbors checked

n_{cell}: total number of single cells

$$F1_{phenotype} = \frac{2 * \left(1 - \text{neigh\_score}_{sample}\right) * (1 - \text{neigh\_score}_{batch}) * \left(\text{neigh\_score}_{phenotype}\right)}{\left(1 - \text{neigh\_score}_{sample}\right) + (1 - \text{neigh\_score}_{batch}) + \left(\text{neigh\_score}_{phenotype}\right)} \quad \text{Equation 2}$$

neigh_score: neighborhood score computed in Equation 1 for a given identity.

APPLICATION OF SCING TO VISIUM SPATIAL TRANSCRIPTOMICS DATA FOR MOUSE AD AND WT BRAIN

To determine the applicability and interpretability of SCING to spatial transcriptomics data, we applied SCING to mouse whole brain AD and WT data.[28] Since the network was built on the whole brain rather than a single cell type, we expect more variance amongst networks from subsamples, therefore we built 1,000 GRNs to be merged into the final network. We partitioned the genes with the Leiden graph partitioning algorithm into 33 modules. Using AUCell from SCENIC,[19] we obtained module specific expression for each spot. We determined regional specificity between pairs of larger regions (cortex, hippocampus, brainstem) through t-tests and overall variance for the smaller subregions through ANOVA. We determine differential module expression between AD and WT through t-tests, and correlation with age or plaque with Pearson correlation. The null distribution of Pearson correlation coefficients was generated by randomly sampling genes with the same number of genes in the module,

computing the AUCell scores for the random gene sets, and computing Pearson correlation between the AUCell scores and the plaque or age. The null distribution of correlation coefficients was used to determine the p value for each module's correlation coefficient. Finally, the module 9, 10, 25, and 30 subnetworks were visualized using Cytoscape[84] and annotated cell type marker genes using the Allen Brain Atlas whole brain Smartseq2 data.[85]