METHOD

Single-cell gene regulatory network inference at scale: The Inferelator 3.0

Claudia Skok Gibbs[1†], Christopher A Jackson[2,3†], Giuseppe-Antonio Saldi[2,3†], Aashna Shah[1], Andreas Tjärnberg[2,3], Aaron Watters[1], Nicholas De Veaux[1], Konstantine Tchourine[6], Ren Yi[4], Tymor Hamamsy[5], Dayanne M Castro[2,3], Nicholas Carriero[7], David Gresham[2,3], Emily R Miraldi[8,9] and Richard Bonneau[1,2,3,4,5*]

*Correspondence: rb133@nyu.edu
[2] Center For Genomics and Systems Biology, NYU, 10008, New York, USA
Full list of author information is available at the end of the article
†These authors contributed equally and are listed alphabetically by last name. They have agreed to change the order of equal contributing authors to list themselves first as a convenience when necessary.

Abstract

Gene regulatory networks define regulatory relationships between transcription factors and target genes within a biological system, and reconstructing them is essential for understanding cellular growth and function. In this work, we present the Inferelator 3.0, which has been significantly updated to integrate data from distinct cell types to learn context-specific regulatory networks and aggregate them into a shared regulatory network, while retaining the functionality of the previous versions. The Inferelator 3.0 reliably learns informative networks from the model organisms *Bacillus subtilis* and *Saccharomyces cerevisiae*. We demonstrate its capabilities by learning networks for multiple distinct neuronal and glial cell types in the developing *Mus musculus* brain at E18 from a large (1.3 million) single-cell gene expression dataset with paired single-cell chromatin accessibility data.

Keywords: Gene Regulation; Network Inference; Transcription Factors; Transcription Factor Activity

1 Background

Gene expression is tightly regulated at multiple levels in order to control growth, development, and response to environmental conditions (Figure 1A). Transcriptional regulation is principally controlled by Transcription Factors (TFs) that bind to DNA and effect chromatin remodeling [1] or directly modulate the output of RNA polymerases [2]. Three percent of Saccharomyces cerevisiae genes are TFs [3], and more than six percent of human genes are believed to be TFs or cofactors [4]. Connections between TFs and genes combine to form a transcriptional Gene Regulatory Network (GRN) that can be represented as a directed graph (Figure 1B). Learning the true regulatory network that connects regulatory TFs to target genes is a key problem in biology [5, 6]. Determining the valid GRN is necessary to explain how mutations that cause gene dysregulation lead to complex disease states [7], how variation at the genetic level leads to selectable phenotypic variation [8, 9], and how to re-engineer organisms to efficiently produce industrial chemicals and enzymes [10].

Learning genome-scale networks relies on genome-wide expression measurements, initially captured with microarray technology [11], but today typically measured by RNA-sequencing (RNA-seq) [12, 13]. A major difficulty is that biological systems

have large numbers of both regulators and targets; there is poor network identifiability because many plausible networks can explain observed expression data and the regulation of gene expression in an organism [14]. Designing experiments to produce data that increases network identifiability is possible [15], but most data is collected for specific projects and repurposed for network inference as a consequence of the cost of data collection. Large-scale experiments in which a perturbation is made and dynamic data is collected over time is exceptionally useful for learning GRNs but systematic studies that collect this data are rare [16].

Measuring the expression of single cells using single-cell RNA-sequencing (scRNA-seq) is an emerging and highly scalable technology. Microfluidic-based single-cell techniques [17, 18, 19] allow for thousands of measurements in a single experiment. Split-pool barcoding techniques [20] are poised to increase single-cell throughput by an order of magnitude. These techniques have been successfully applied to generate multiplexed gene expression data from pools of barcoded cell lines with loss-of-function TF mutants [21, 22], enhancer perturbations [23], and disease-causing oncogene variants [24]. Individual cell measurements are sparser and noisier than measurements generated using traditional RNA-seq, although in aggregate the gene expression profiles of single-cell data match RNA-seq data well [25], and techniques to denoise single-cell data have been developed [26, 27].

The seurat [28] and scanpy [29] bioinformatics toolkits are established tools for single-cell data analysis, but pipelines for inferring GRNs from single-cell data are still nascent. The SCENIC [30] GRN inference pipeline is based around the GENIE3 method that uses ensemble regression trees [31] to estimate the importance of TFs in explaining gene expression profiles. CellOracle [32] has been recently proposed as a pipeline to integrate single-cell ATAC and expression data using a motif-based search for potential regulators, followed by Bayesian ridge regression to enforce sparsity in the output GRN. SCODE [33] infers GRNs by solving linear ordinary differential equations using time-course single-cell data. GRN inference is computationally challenging, and the most scalable of these GRN pipelines has learned GRNs from 50,000 cells of gene expression data [30].

Here we describe the Inferelator 3.0 pipeline for single-cell GRN inference, based on regularized linear regression [34]. The Inferelator 2.0 [35] has performed well inferring networks from *Bacillus subtilis* [36], human Th17 cells [37, 38], mouse Lymphocytes [39], *Saccharomyces cerevisiae* [40], and *Oryza sativa* [41]. We have re-implemented the Inferelator 3.0 with new functionality in python to learn GRNs from single-cell gene expression data. Specifically, this new package provides scalability, allowing millions of cells to be analyzed together, as well as integrated support for multi-task GRN inference, while retaining the ability to utilize bulk gene expression data. As a demonstration of these capabilities, we learn GRNs from several model organisms, and generate a mouse neuronal GRN from a publicly available dataset containing 1.3 million cells.

2 Results

2.1 The Inferelator 3.0

In the 11 years since the last major release of the Inferelator [35], the scale of data collection in biology has accelerated enormously. We have therefore rewritten the

Inferelator as a python package to take advantage of the concurrent advances in data processing. For inference from small scale gene expression datasets ($< 10^4$ observations), the Inferelator 3.0 uses native python multiprocessing to run on individual computers . For inference from extremely large scale gene expression datasets (10^4 - 10^7 observations) that are increasingly available from scRNA-seq experiments, the Inferelator 3.0 takes advantage of the Dask analytic engine [42] for deployment to high-performance clusters (Figure 1C), or for deployment as a kubernetes image to the Google cloud computing infrastructure.

2.2 Network Inference using Bulk RNA-Seq Expression Data

We have incorporated several network inference model selection methods into the Inferelator (Figure 2A). In order to evaluate the network inference performance of these methods, we test on the prokaryotic model *Bacillus subtilis* and the eukaryotic model *Saccharomyces cerevisiae*. Both *B. subtilis* [36, 43] and *S. cerevisiae* [40, 16] have large bulk RNA-seq and microarray gene expression datasets, in addition to a relatively large number of experimentally determined TF-target gene interactions that can be used as a gold standard for assessing network inference.

Using two independent datasets for each organism, we find that the model selection methods Bayesian Best Subset Regression (BBSR) [44] and Stability Approach to Regularization Selection for Least Absolute Shrinkage and Selection Operator (StARS-LASSO) [38] perform equivalently (Figure 2B). The datasets separate on the first principal component (Supplemental Figure 1A), indicating that there are substantial batch-specific effects between these independent datasets. These dataset-specific batch effects make combining this data for network inference difficult; conceptually, each dataset is in a separate space, and must be mapped into a shared space if they are to be combined. We take a different approach to addressing the batch effects between datasets by treating them as separate learning tasks [45] and then combining network information into a unified GRN. This results in a considerable improvement in network inference performance over either dataset individually (Figure 2C). The best performance is obtained with Adaptive Multiple Sparse Regression (AMuSR) [45], a multi-task learning method that shares information between tasks during regression. The GRN learned with AMuSR explains the variance in the expression data better than learning networks from each dataset individually with BBSR or StARS-LASSO and then combining them (Supplemental Figure 1B). There is a high overlap in the number of GRN edges learned from each dataset, showing that there is a common network across different tasks (Supplemental Figure 1C).

2.3 Generating Prior Networks from Chromatin Data and Transcription Factor Motifs

The Inferelator produces an inferred network from a combination of gene expression data and a prior network constructed from existing knowledge about known gene regulation. Curated databases of regulator-gene interactions culled from domain-specific literature are an excellent source for prior networks. While some model systems have excellent databases of known interactions, these resources are unavailable for most organisms or cell types. In these cases, using chromatin accessibility determined by a standard Assay for Transposase-Accessible Chromatin (ATAC) in

combination with the known DNA-binding preferences for TFs to identify putative target genes is a viable alternative [38].

To generate these prior networks we have developed the inferelator-prior accessory package that uses TF motif position-weight matrices to score TF binding within gene regulatory regions and build sparse prior networks (Figure 3A). These gene regulatory regions can be identified by ATAC, by existing knowledge from TF Chromatin Immunoprecipitation (ChIP) experiments, or from known databases (e.g. ENCODE [46]). Here, we compare the inferelator-prior tool to the CellOracle package [32] that also constructs motif-based networks that can be constrained to regulatory regions, in *Saccharomyces cerevisiae* by using sequences 200bp upstream and 50bp downstream of each gene TSS as the gene regulatory region. The inferelator-prior and CellOracle methods produce networks that are similar when measured by Jaccard index but are dissimilar to the YEASTRACT literature-derived network (Figure 3B). These motif-derived prior networks from both the inferelator-prior and CellOracle methods perform well as prior knowledge for GRN inference using the Inferelator pipeline (Figure 3C). The source of the motif library has a significant effect on network output, as can be seen with the well-characterized TF GAL4. GAL4 has a canonical $CGGN_{11}CGG$ binding site; different motif libraries have different annotated binding sites (Supplemental Figure 2A) and yield different motif-derived networks with the inferelator-prior pipeline (Supplemental Figure 2B-C).

2.4 Network Inference using Single-Cell Expression Data

Single-cell data is undersampled and noisy, but large numbers of observations are collected in parallel and count data derived from unique molecular identifiers have some intrinsic advantages. In order to quantitatively evaluate network inference performance, we apply the Inferelator to *Saccharomyces cerevisiae* single-cell expression data [22, 47], and score the model performance based on a previously-defined yeast gold standard [40]. This data is split into 15 separate tasks, based on labels that correspond to experimental conditions from the original works (Figure 4A). A network is learned for each task separately using the YEASTRACT literature-derived prior, and aggregated into a final network for scoring on held-out genes from the gold standard. We test a combination of several preprocessing options with three network inference model selection methods (Figure 4B-D).

We find that network inference is generally sensitive to the preprocessing options chosen, and that these differences due to preprocessing generally outweigh the differences between different model selection methods (Figure 4B-D). A standard Freeman-Tukey or $\log_2$ pseudocount transformation on raw count data yields the best performance, with notable decreases in recovery of the gold standard when count data is depth-normalized (such that each cell has the same total transcript counts). The performance of the randomly generated Noise control (N) is higher than the performance of the shuffled (S) control when counts per cell are not normalized, suggesting that total counts per cell provides additional information during inference.

Different model performance metrics, like AUPR, Matthews Correlation Coefficient (MCC), and F1 score correlate very well and identify the same optimal hyperparameters (Supplemental Figure 3). We apply StARS-LASSO to data that

has been Freeman-Tukey transformed to generate a final network without holding out genes for cross-validation (Figure 4E). While we use AUPR as a metric for evaluating model performance, selecting a threshold for including edges in a GRN by precision or recall requires a target precision or recall to be chosen arbitrarily. Alternatively, MCC and F1 score allow a threshold to be determined that maximizes similarity to the known prior or gold standard GRN. Choosing the Inferelator confidence score threshold to include edges in a final network that maximizes MCC is a simple heuristic way to select the size of a learned network that maximizes the overlap of the gold standard while minimizing links not in the gold standard (Figure 4F and section 5.6). Using the maximum F1 score is also an option, but gives a less conservative GRN as true negatives are not considered and will not diminish the score. Both metrics balance similarity to the gold standard with overall network size, and therefore represent straightforward heuristics that do not rely on arbitrary thresholds.

2.5 Large-scale Single-Cell Mouse Neuron Network Inference

To show scalability, we apply the Inferelator to a large-scale (1.3 million cells of scRNA-seq expression data) publicly available dataset of mouse brain cells (10x genomics) that is accompanied by 15,000 single-cell ATAC (scATAC) measurements. By using Dask to parallelize network inference, we are able to distribute work across multiple computational nodes, allowing networks to be rapidly learned from ¿10^5 cells (Figure 4A). We separate the expression and scATAC data into broad categories; Excitatory neurons, Interneurons, Glial cells and Vascular cells (Figure 5A-E). After initial quality control, filtering, and cell type assignment, 766,402 scRNA-seq and 7,751 scATAC observations remain (Figure 5F, Supplemental Figure 4B-D). scRNA-seq data is further clustered within broad categories into clusters (Figure 5B) that are assigned to specific cell types based on marker expression (Figure 5C, Supplemental Figure 5). scATAC data is aggregated into chromatin accessibility profiles for Excitatory neurons, Interneurons, and Glial cells (Figure 5D) based on accessibility profiles (Figure 5E), which are then used with the TRANSFAC mouse motif position-weight matrices to construct prior knowledge networks with the inferelator-prior pipeline. Most scRNA-seq cell type clusters have thousands of cells, however a few clusters of rare cell types have as few as 42 (Figure 5G)

After processing scRNA-seq into 36 cell type clusters and scATAC data into 3 broad (Excitatory neurons, Interneurons, and Glial) priors, we used the Inferelator to learn an aggregate mouse brain GRN. Each of the 36 clusters was assigned the most appropriate of the three prior networks and learned as a separate task using the AMuSR model selection framework. The resulting aggregate network contains 20,991 TF - gene regulatory edges, selected from the highest confidence predictions to maximize MCC (Figure 6A-B). 1,909 of these network edges are present in every task-specific network, implying that they are a common regulatory core (Figure 6C). Task-specific networks from similar cell types tend to be highly similar, as measured by Jaccard index (Figure 6D). We learn very similar GRNs from each excitatory neuron task, and very similar GRNs from each interneuron task, although each of these broad categories yields different regulatory networks. There are also notable examples where glial and vascular tasks produce GRNs that are distinctively

different from other glial and vascular GRNs. Finally, we can examine specific TFs and compare networks between cell type categories (Supplemental Figure 6). The TFs Egr1 and Atf4 are expressed in all cell types and Egr1 is known to have an active role at embryonic day 18 (E18) [48]. In our learned network, Egr1 targets 103 genes, of which 20 are other TFs (Figure 6E-G). Half of these targets (49) are common to both neurons and glial cells, while 38 target genes are specific to neuronal GRNs and 16 target genes are specific to glial GRNs. We identify 14 targets for Atf4 (Figure 6H), the majority of which (8) are common to both neurons and glial cells, with only 1 target gene specific only to neuronal GRNs and 5 targets specific only to glial GRNs.

3 Discussion

We have developed the Inferelator v3.0 software package to address several key needs in single-cell gene regulatory network inference that have remained difficult to meet with existing solutions. First, this package is well-documented and straightforward to install and run on an individual computer, in the cloud, or on a large cluster. The Inferelator workflow can be scaled to match the size of the network inference problem and has no organism-specific requirements that preclude easy application to non-standard organisms. Second, different model selection methods can be compared with identical pre- and post-processing methods, including arbitrary methods implemented through the common scikit-learn estimator framework. Model baselines can be easily established by setting flags to shuffle labels or generate noised data sets, and cross-validation and scoring on holdout genes is built directly into the pipeline. We believe this is particularly important, as many of the performance differences between gene regulatory network inference methods are not due to clever methods for model selection, but are instead the result of differences in data cleaning and preprocessing. Third, we have suggested a principled method for selecting regulatory edges to retain in a GRN. Many GRNs have been inferred by applying a collection of arbitrary heuristics to potential regulatory edges; here we propose ranking regulatory edges by the amount of target gene variance that they explain, and then selecting a threshold for inclusion that maximizes the MCC when scored against a known prior or gold standard network. Finally, we have evaluated the network inference performance on several model organisms that have been well-studied, and for which a reasonable gold standard ground truth GRN can be created from experimental literature. Complex eukaryotes (e.g. mice or humans) lack a gold standard ground truth that can be used to determine real-world network inference performance. Many GRN inference methods instead benchmark on simulated or toy data, with limited experimental validation of a carefully selected tiny subset of an inferred real-world GRNs, making method comparisons difficult.

Multi-task modeling is also a key advantage for single-cell GRN inference. Unlike traditional RNA-seq that effectively measures the average gene expression of large number of cells, scRNA-seq can yield individual measurements for many different cell types that are implementing distinct regulatory programs. Learning GRNs from each of these cell types as a separate learning task in a multi-task framework allows cell type differences to be retained, while still taking advantage of the common regulatory programs. We demonstrate the use of this multi-task approach to simultaneously learn regulatory GRNs for a variety of mouse neuronal cell types from a

very large (10^6) single-cell data set. This includes learning GRNs for rare cell types; by sharing information between cell types during regression, we are able to learn a core regulatory network while also retaining cell type specific interactions. As an example, the TFs Egr1 and Atf4 are broadly expressed and have multiple functions from memory formation and post synaptic development in neurons to cell migration and genome methylation in many cell types [48, 49]. We find a number of target genes regulated by Egr1 across all neuronal and glial cell types, like the RNA-binding protein Nova2 that regulates alternative splicing and axonal development [50] (Figure 6F). The stress response TF Atf4 [51] is known to regulate neuronal $GABA_B$ receptor trafficking [52], and we identify it as a regulator of Rnf166, a RING-finger protein that promotes apoptotic cell death in neurons [53]. We also determine that Atf4 regulates the highly conserved cytosolic sulfotransferase Sult4a1, which modulates neuronal branching complexity and dendritic spine formation, and has been linked to neurodevelopmental disorders [54]. As the GRNs that have been learned for each cell type are sparse and consist of the highest-confidence regulatory edges, they are very amenable to exploration and experimental validation.

A number of limitations remain that impact our ability to accurately predict gene expression and cell states. Most important is a disconnect between the linear modeling that we use to learn GRNs and the non-linear biophysical models that incorporate both transcription and RNA decay. Modeling strategies that more accurately reflect the underlying biology will improve GRN inference directly, and will also allow prediction of useful latent parameters (e.g. RNA half-life) that are experimentally difficult to access. It is also difficult to determine if regulators are activating or repressing specific genes [32], complicated further by biological complexity that allows TFs to switch between activation and repression [55]. Improving prediction of the directionality of network edges, and if directionality is stable in different contexts would also be a major advance. Many TFs bind cooperatively as protein complexes, or antagonistically via competitive binding, and explicit modeling of these TF-TF interactions would also improve GRN inference and make novel biological predictions. Finally, we note that core regulatory networks are likely to be conserved between related species, and further work to develop multi-species inference techniques can leverage evolutionary relationships to improve GRN inference [56]. The modular Inferelator 3.0 framework will allow us to further explore these open problems in regulatory network inference without having to repeatedly reinvent and reimplement existing work.

4 Conclusion

The Inferelator 3.0 is a state-of-the-art, easily deployable, and highly scalable network inference tool that is generally applicable to learning GRNs from both single-cell and traditional RNA-seq experiments in any organism of interest. With its accessory software packages, genome-wide expression data of any type can be integrated with chromatin accessibility data to construct and explore cell type-specific GRNs. We have established the reliability of this tool by benchmarking on real-world data in model organisms *Bacillus subtilis* and *Saccharomyces cerevisiae* with known gold standard GRNs, and demonstrated how it could be applied to large-scale network inference on many different cell types in the developing mouse brain.

We expect this to be a valuable tool to build biologically-relevant GRNs for experimental follow-up, as well as a baseline for further development of computational methods in the network inference field.

5 Methods

5.1 TF Motif-Based Connectivity Matrix (inferelator-prior)

A prior knowledge matrix consists of a signed or unsigned connectivity matrix between regulatory transcription factors (TFs) and target genes. This matrix can be obtained experimentally or by mining regulatory databases. Scanning genomic sequence near promoter regions for TF motifs allows for the construction of motif-derived priors which can be further constrained experimentally by incorporating information about chromatin accessibility [38].

We have further refined the generation of prior knowledge matrices with the python inferelator-prior package, which takes as input a gene annotation GTF file, a genomic FASTA file, and a TF motif file, and generates an unsigned connectivity matrix. It has dependencies on the common scientific computing packages NumPy [57], SciPy [58], and scikit-learn [59]. In addition, it uses the BEDTools kit [60] and associated python interface pybedtools [61]. The inferelator-prior package (v0.3.0 was used to generate the networks in this manuscript) is available on github (`https://github.com/flatironinstitute/inferelator-prior`) and can be installed through the python package manager pip.

5.1.1 Motif Databases

DNA binding motifs were obtained from published databases. CISBP [62] motifs were obtained from CIS-BP (`http://cisbp.ccbr.utoronto.ca/`; Build 2.00; Downloaded 11/25/2020) and processed into a MEME-format file with the PWMtoMEME module of inferelator-prior. JASPAR [63] motifs were obtained as MEME files from JASPAR (`http://jaspar.genereg.net/`; 8th Release; Downloaded 11/25/2020) . TRANSFAC [64] motifs were licensed from geneXplain (`http://genexplain.com/transfac/`; Version 2020.1; Downloaded 09/13/2020) and processed into a MEME-format file with the inferelator-prior motif parsing tools. A network of literature-curated network edges was obtained as a gold standard from the YEASTRACT database [65, 66] (`http://www.yeastract.com/`; Downloaded 07/13/2019).

5.1.2 Motif Scanning

Genomic regions of interest are identified by locating annotated Transcription Start Sites (TSS) and opening a window that is appropriate for the organism. For microbial species with a compact genome (e.g. yeast), regions of interest are defined as 1000bp upstream and 100bp downstream of the TSS. For complex eukaryotes with large intergenic regions (e.g. mammals), regions of interest are defined as 50000bp upstream and 2500bp downstream of the TSS. This is further constrained by intersecting the genomic regions of interest with a user-provided BED file, which can be derived from a chromatin accessibility experiment (ATAC-seq) or any other method of identifying chromatin of interest. Within these regions of interest, motif locations are identified using the Find Original Motif Occurrences (FIMO) [67] tool from the

MEME suite [68], called in parallel on motif chunks to speed up processing. Each motif hit identified by FIMO is then scored for information content (IC) [69]. IC_i, ranging between 0 and 2 bits, is calculated for each base i in the binding site, where $p_{b,i}$ is the probability of the base b at position i of the motif and $p_{b,bg}$ is the background probability of base b in the genome (Equation 1). Effective information content (EIC) (Equation 2) is the sum of all motif at position i is IC_i penalized with the ℓ_2-norm of the hit IC_i and the consensus motif base at position i, $\mathrm{IC}_{i,\mathrm{consensus}}$.

$$\mathrm{IC}_i = p_{b,i} \log_2 \left(\frac{p_{b,i}}{p_{b,bg}} \right) \tag{1}$$

$$\mathrm{EIC} = \sum_i \mathrm{IC}_i - |\, \mathrm{IC}_i - \mathrm{IC}_{i,\mathrm{consensus}} \,|_2^2 \tag{2}$$

5.1.3 Connectivity Matrix

A TF-gene binding score is calculated separately for each TF and gene. Each motif hit for a TF within the region of interest around the gene is identified. Overlapping motif hits are resolved by taking the maximum IC for each overlapping base, penalized with the ℓ_2-norm of differences from the motif consensus sequence. To account for cooperative TF binding effects, any motif hits within 100 bases (25 bases for yeast) are combined, and their EIC scores are summed. The TF-gene binding score is the maximum TF EIC after accounting for overlapping and adjacent TF motifs, and all TF-gene scores are assembled into a Genes x TFs score matrix.

This unfiltered TF-gene score matrix is not sparse as motifs for many TFs are expected to occur often by chance, and TF-gene scores for each TF are not comparable to scores for other TFs as motif position-weight matrices have differing information content. Scores for each TF are clustered using the density-based k-nearest neighbors algorithm DBSCAN [70] (MinPts = 0.001 * number of genes, eps = 1). The cluster of TF-gene edges with the highest score values, and any high-score outliers, are retained in the connectivity matrix, and other TF-gene edges are discarded.

5.1.4 CellOracle Connectivity Matrix

CellOracle [32] was cloned from github (v0.6.5; `https://github.com/morris-lab/CellOracle`; a0da790). CellOracle was provided a BED file with promoter locations for each gene (200bp upstream of transcription start site to 50bp downstream of transcription start site) and the appropriate MEME file for each motif database. Connectivity matrices were predicted using a false positive rate of 0.02 and a motif score threshold of 6. The inferelator-prior pipeline was run using the same promoter locations and MEME files so that the resulting networks are directly comparable, and the Jaccard index between each network and the YEASTRACT network was calculated. Each motif-based network was used as a prior for inferelator network inference on *Saccharomyces cerevisiae*, with the same 2577 genome-wide expression microarray measurements [40]. 20% of the genes were held out of the prior networks and used for scoring the resulting network inference. The motif-based network files have been included in Supplemental Data 1.

5.2 Network Inference (The Inferelator)

The Inferelator modeling of gene regulatory networks relies on three main modeling assumptions. First, because many transcription factors (TFs) are post transcriptionally controlled and their expression level may not reflect their underlying biological activity, we assume that the activity of a TF can be estimated using expression levels of known targets from prior interactions data [36, 71]. Second, we assume that gene expression can be modeled as a weighted sum of the activities of TFs [34, 45]. Finally, we assume that each gene is regulated by a small subset of TFs and regularize the linear model to enforce sparsity.

The Inferelator was initially developed and distributed as an R package [34, 44, 72, 73]. We have rewritten it as a python package with dependencies on the common scientific computing packages NumPy [57], SciPy [58], pandas [74], AnnData [29], and scikit-learn [59]. Scaling is implemented either locally through python or as a distributed computation with the Dask [42] parallelization library. The inferelator package (v0.5.4 was used to generate the networks in this manuscript) is available on github (`https://github.com/flatironinstitute/inferelator`) and can be installed through the python package manager pip. The Inferelator takes as input gene expression data and prior information on network structure, and outputs ranked regulatory hypotheses of the relative strength and direction of each interaction with an associated confidence score.

5.3 Transcription Factor Activity

The expression level of a TF is often not suitable to describe its activity [75]. Transcription factor activity (TFA) is an estimate of the latent activity of a TF that is inducing or repressing transcription of its targets in a sample. A gene expression dataset ($\mathbf{X}$) is a matrix where $X_{i,j}$ is the observed mRNA expression level ($i \in$ Samples and $j \in$ Genes), measured either by microarray, RNA-seq, or single cell RNA sequencing (scRNA-seq).

$$X_{i,j} = \sum_k A_{i,k} P_{k,j} \tag{3}$$

We estimate TFA by solving (Equation 3) for activity ($A_{i,k}$), where $k \in$ TFs, and P is a prior connectivity matrix where $P_{k,j}$ is 1 if gene j is regulated by TF k and 0 if it is not. In matrix notation, $\mathbf{X} = \mathbf{AP}$, and $\hat{\mathbf{A}}$ is estimated by minimizing $\| \hat{\mathbf{A}}\mathbf{P} - \mathbf{X} \|_2^2$. This is calculated by taking the pseudoinverse of $\mathbf{P}$ and solving $\hat{\mathbf{A}} = \mathbf{X}\mathbf{P}^{-1}$. The resulting $\hat{\mathbf{A}}$ is a matrix where $\hat{A}_{i,k}$ is the estimated latent TFA for sample i and TF k. In cases where all values in $\mathbf{P}$ for a TF are 0, that TF is removed from $\mathbf{P}$ and the expression $\mathbf{X}$ of that TF is used in place of activity.

5.4 Inferelator Network Inference

Linear models (Equation 4) are separately constructed for each gene j.

$$X_i = \sum_k \hat{A}_{i,k} \beta_k \tag{4}$$

In addition to the model selection methods described here, we have implemented a module which takes any scikit-learn regression object (for example, elastic net [76]). Model selection and regularization techniques are applied to enforce the biological property of sparsity. If the coefficient $\beta_{j,k}$ is non-zero, it is evidence for a regulatory relationship between TF k and gene j.

$$S_{j,k} = 1 - \frac{\sigma^2_{allTFs}}{\sigma^2_{TF_k leaveout}} \tag{5}$$

For each gene j, the amount of variance explained by each regulatory TF k is calculated as the ratio between the variance of the residuals in the full model and the variance of the residuals when the linear model is refit by ordinary least squares (OLS) and k is left out (Equation 5).

In order to mitigate the effect of outliers and sampling error, model selection is repeated multiple times using input expression data $\mathbf{X}$ that has been bootstrapped (resampled with replacement). Predicted TF-gene interactions are ranked for each bootstrap by amount of variance explained and then rank-combined into a unified network prediction. Confidence scores are assigned based on the combined rank for each interaction, and the overall network is compared to a gold standard and performance is evaluated by area under the precision-recall curve.

The effects of setting hyperparameters can be tested by cross-validation on the prior and gold standard networks. This strategy holds out a subset of genes (rows) from the prior knowledge network $\mathbf{P}$. Network inference performance is then evaluated on only those held-out genes, using the gold standard network.

5.4.1 Model Selection: Bayesian Best Subset Regression

Bayesian Best Subset Regression (BBSR) is a model selection method described in detail in [73]. Initial feature selection for this method is necessary as best subset regression on all possible combinations of hundreds of TF features is computationally intractible. We therefore select ten TF features with the highest context likelihood of relatedness between expression of each gene and activity of each TF. This method is described in detail in [72].

First, gene expression and TF activity are discretized into equal-width bins (n=10) and mutual information is calculated based on their discrete probability distributions (Equation 6) to create a mutual information matrix $\mathbf{M^{dyn}}$.

$$M^{dyn}_{j,k} = p(X_j, \hat{A}_k) \log \frac{p(X_j, \hat{A}_k)}{p(X_j)p(\hat{A}_k)} \tag{6}$$

$$M^{stat}_{k_1,k_2} = p(\hat{A}_{k_1}, \hat{A}_{k_2}) \log \frac{p(\hat{A}_{k_1}, \hat{A}_{k_2})}{p(\hat{A}_{k_1})p(\hat{A}_{k_2})} \tag{7}$$

Mutual information is also calculated between activity of each TF (Equation 7) to create a mutual information matrix $\mathbf{M^{stat}}$.

$$z_{j,k}^{dyn} = \frac{Mj, k^{dyn} - \sum_j \frac{Mj,k^{dyn}}{n_i}}{\sigma_k^{dyn}} \tag{8}$$

$$z_{j,k}^{stat} = \frac{Mj, k^{dyn} - \sum_j \frac{Mj,k^{stat}}{n_i}}{\sigma_k^{stat}} \tag{9}$$

$$z_{j,k}^{mixed} = \sqrt{(z_{j,k}^{dyn})^2 + (z_{j,k}^{stat})^2} \tag{10}$$

A mixed context likelihood of relatedness score is then calculated as a pseudo-zscore by calculating $\mathbf{Z^{dyn}}$ (Equation 8) and $\mathbf{Z^{stat}}$ (Equation 9). Any values less than 0 in $\mathbf{Z^{dyn}}$ or $\mathbf{Z^{stat}}$ are set to 0, and then they are combined into a mixed context likelihood of relatedness matrix $\mathbf{Z^{mixed}}$ (Equation 10). For each gene j, the 10 TFs with the highest mixed context likelihood of relatedness values are selected for regression.

For best subset regression, a linear model is fit with OLS for every combination of the selected predictor variables.

$$\rho(\beta, \sigma^2|X_j) = \rho(\beta|X_j, \sigma^2)\rho(\sigma^2|X_i) \tag{11}$$

$$\rho(\sigma^2|X_i) \propto IG(\frac{n}{2}, \frac{SSR}{2} + \frac{(\beta_0 - \beta_{\text{OLS}})\mathbf{GX'XG}(\beta_0 - \beta_{\text{OLS}})}{2}) \tag{12}$$

We define β_0 as our null prior for the model parameters (zeros), β_{OLS} as the model coefficients from OLS, SSR as the sum of squared residuals, and $\mathbf{G}$ as a g-prior diagonal matrix where the diagonal values represent a weight for each predictor variable. g-prior weights in $\mathbf{G}$ close to 0 favor β values close to β_0. Large g-prior weights favor β values close to β_{OLS}. By default, we select g-prior weights of 1 for all predictor variables. From the joint posterior distribution (Equation 11) we can calculate the marginal posterior distribution of σ^2 (Equation 12), where IG is the inverse gamma distribution. The Bayesian information criterion (BIC) is calculated for each model, where n is the number of observations and k is the number of predictors (Equation 13).

$$BIC = n \ln(\sigma^2) - k \ln(n) \tag{13}$$

$$E[\sigma^2] = \frac{\frac{SSR}{2} + \frac{(\beta_0 - \beta_{\text{OLS}})\mathbf{GX'XG}(\beta_0 - \beta_{\text{OLS}})}{2}}{\frac{n}{2} - 1} \tag{14}$$

$$E[BIC] = n(\ln(\frac{SSR}{2} + \frac{(\beta_0 - \beta_{\text{OLS}})\mathbf{GX'XG}(\beta_0 - \beta_{\text{OLS}})}{2}) - Digamma(\frac{n}{2})) - k \ln(n) \tag{15}$$

We calculate the expected posterior distribution of σ^2 (Equation 14) for each subset of predictors, and use it to determine the model BIC (Equation 15). We then select the model with the smallest $E[BIC]$. The predictors in the selected subset model for gene j are TFs which regulate its expression.

5.4.2 Model Selection: StARS-LASSO

Least absolute shrinkage and selection operator (LASSO) [77] combined with the Stability Approach to Regularization Selection (StARS) [78] is a model selection method described in detail in [38]. In short, the StARS-LASSO approach is to select the optimal λ parameter for (Equation 16). N random subsamples of X and $\hat{A}$ without replacement subnetworks $S_{n,\lambda}$ are defined as the non-zero coefficients $\beta_{n,\lambda}$ after LASSO regression. Initially, λ is set large, so that each subnetwork S_n is highly sparse, and is then decreased, resulting in increasingly dense networks. Edge instability is calculated as the fraction of times subnetworks disagree about the presence of an network edge. As λ decreases, the subnetworks are expected to have increasing edge instability initially and then decreasing edge instability as λ approaches 0, as (Equation 16) reduces to OLS and each subnetwork becomes dense.

$$\min_{\beta} \frac{1}{2n} |X - \hat{A}\beta|_2^2 - \lambda|\beta|_1 \tag{16}$$

We choose the largest value of λ such that the edge instability is less than 0.05, which is interpretable as all subnetworks share $> 95\%$ of edges. This selection represents a balance between increasing the network size and minimizing the instability that occurs when data is sampled.

5.5 Multiple Task Network Inference

We separate biological samples which represent different states into separate tasks, learn networks from these tasks, and then combine task-specific networks into an ensemble network. One method of solving these states is to sequentially apply a single-task method for network inference (i.e. 5.4.1 or 5.4.2). The networks generated for each task are then rank-combined into a unified network. The Adaptive Multiple Sparse Regression (AMuSR) method, described in detail in [45], uses a multi-task learning framework, where each task is solved together.

$$\arg\min_{B,S} \frac{1}{2n} \|X_{d,i} - (S_d + B)\hat{A}_d\|_2^2 + \lambda_s \|S\|_{1,1} + \lambda_b \|B\|_{1,\infty} \tag{17}$$

$$\hat{W} = \hat{B} + \hat{S} \tag{18}$$

In (Equation 17), **B** is a block-sparse weight matrix in which the weights for any feature are the same across all tasks. **S** is a sparse weight matrix in which the weights for features can vary between tasks. The combination **W** of **B** and **S** (Equation 18) are model weights representing regulatory interactions between TFs and genes. In

short, this method uses adaptive penalties to favor regulatory interactions shared across multiple tasks in $\mathbf{B}$, while recognizing dataset specific interactions in $\mathbf{S}$. Model hyperparameters λ_s and λ_b are identified by grid search, selecting the model that minimizes the extended Bayesian Information Criterion (eBIC) (Equation 19), where D is the number of task datasets, and for dataset d, n_d is the number of observations, $X_i^{(d)}$ is gene expression for gene i, $\hat{A}^{(d)}$ is TF activity estimates, $W_{*,d}$ is model weights, k_d is the number of non-zero predictors, and p_d is the total number of predictors. For this work, we choose to set the eBIC paramater γ to 1.

$$eBIC = \frac{1}{D}\sum n_d \ln \frac{1}{n_d}\|X_i^{(d)} - \hat{A}^{(d)T}W_{*,d}\|_2^2 + k_d \ln n_d + 2\gamma \ln \binom{p_d}{k_d} \tag{19}$$

5.6 Network Performance Metrics

Prior work has used the area under the Precision (Equation 20) - Recall (Equation 21) curve to determine performance, by comparing to some known, gold-standard network. Here we add two metrics; Matthews correlation coefficient [79] (MCC) (Equation 22) and F1 score (Equation 23). MCC can be calculated directly from the confusion matrix True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) values.

$$Precision = \frac{TP}{TP + FP} \tag{20}$$

$$Recall = \frac{TP}{TP + FN} \tag{21}$$

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{22}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{23}$$

We compute an MCC and F1 score for each cutoff along ranked interactions in order to generate MCC and F1 scores for all possible networks in growing ranked order. The maximum MCC along ranked interactions gives the subnetwork that has maximum similarity to the comparison network, accounting for TP, FP, TN, and FN. The maximum F1 along ranked interactions gives the subnetwork that has maximum similarity to the comparison network accounting for TP, FP, and FN.

5.7 Network Inference in *Bacillus subtilis*

Microarray expression data for *Bacillus subtilis* was obtained from NCBI GEO; GSE67023 [36] (n=268) and GSE27219 [43] (n=266). The Inferelator (v0.5.4) learned GRNs using each expression dataset separately in conjunction with a known prior network [36] (Supplemental Data 1). Performance was evaluated by AUPR on ten replicates by holding 20% of the genes in the known prior network out, learning the GRN, and then scoring based on the held-out genes. Baseline shuffled controls were performed by randomly shuffling the labels on the known prior network.

Multi-task network inference uses the same *B. subtilis* prior for both tasks, with 20% of genes held out for scoring. Individual task networks are learned and rank-combined into an aggregate network. Performance was evaluated by AUPR on the held-out genes.

5.8 Network Inference in *Saccharomyces cerevisiae*

A large microarray dataset was obtained from NCBI GEO and normalized for a previous publication [40] (n=2,577). It is available on zenodo with DOI: 10.5281/zenodo.3247754. In short, this data was preprocessed with limma [80] and quantile normalized. A second microarray dataset consisting of a large dynamic perturbation screen [16] was obtained from NCBI GEO accession GSE142864 (n=1,693). This dataset is log_2 fold change of an experimental channel over a control channel which is the same for all observations. The Inferelator (v0.5.4) learned GRNs using each expression dataset separately in conjunction with a known YEASTRACT prior network [65, 66] (Supplemental Data 1). Performance was evaluated by AUPR on ten replicates by holding 20% of the genes in the known prior network out, learning the GRN, and then scoring based on the held-out genes in a separate gold standard [40]. Baseline shuffled controls were performed by randomly shuffling the labels on the known prior network.

Multi-task network inference uses the same YEASTRACT prior for both tasks, with 20% of genes held out for scoring. Individual task networks are learned and rank-combined into an aggregate network, which is then evaluated by AUPR on the held-out genes in the separate gold standard.

5.9 Single-Cell Network Inference in *Saccharomyces cerevisiae*

Single-cell expression data for *Saccharomyces cerevisiae* was obtained from NCBI GEO (GSE125162 [22] and GSE144820 [47]). Individual cells (n=44,343) are organized into one of 14 groups based on experimental metadata and used as separate tasks in network inference. Genes were filtered such that any gene with fewer than than 2217 total counts in all cells (1 count per 20 cells) was removed. Data was used as raw, unmodified counts, was Freeman-Tukey transformed ($\sqrt{x+1}+\sqrt{x}-1$), or was $\log_2$ pseudocount transformed ($\log_2(x+1)$). Data was either not normalized, or depth normalized by scaling so that the sum of all counts for each cell is equal to the median of the sum of counts of all cells. For each set of parameters, network inference is run 10 times, using the YEASTRACT network as prior knowledge with 20% of genes held out for scoring. For noise-only controls, gene expression counts are simulated randomly such that for each gene i, $x_i \sim N(\mu_{x_i}, \sigma_{x_i})$ and the sum for each cell is equal to the sum in the observed data. For shuffled controls, the gene labels on the prior knowledge network are randomly shuffled.

5.10 Single-Cell Network Inference in *Mus musculus* neurons

Single-cell expression data from *Mus musculus* brain samples taken at E18 was obtained from 10x genomics [81]. SCANPY was used to preprocess and cluster the scRNAseq dataset. Genes present in fewer than 2% of cells were removed. Cells were filtered out when fewer than 1000 genes were detected, the cell had more than 20,000 total gene counts, or the cell had more than 7% of gene counts assigned to

mitochondrial transcripts. Transcript counts were then log transformed and normalized and scaled. Cells were assigned to mitotic or post mitotic phase based on cell cycle marker genes using score_genes_cell_cycle [82]. In order to focus on neuronal cells, all 374,369 mitotic cells were removed. Remaining cells were clustered by Leiden clustering (Resolution = 0.5) using the first 300 principal components of the 2000 most highly variable genes. Broad cell types were assigned to each cluster based on the expression of marker genes Neurod6 for Excitatory neurons, Gad1 for Interneurons, and Apoe for glial cells. Cells from each broad cell type were then re-clustered into clusters based on the 2000 most highly variable genes within the cluster. Specific cell types were assigned to each subcluster based on the expression of marker genes[83]. Ambiguous clusters were discarded, removing 151,765 cells, leaving resulting in 36 specific cell type clusters that consist of 766,402 total cells.

Single-cell ATAC data from *Mus musculus* brain samples taken at E18 was obtained from 10x genomics; datasets are from samples prepared fresh [84], samples dissociated and cryopreserved [85], and samples flash-frozen [86]. ChromA [87] and SnapATAC [88] were used to process the scATACseq datasets. Consensus peaks were called on the 3 datasets using ChromA. Each dataset was then run through the SnapATAC pipeline using the consensus peaks. Cells were clustered and labels from the scRNAseq object were transferred to the scATAC data. Cells that did not have an assignment score $\geq .5$ were discarded. Assigned barcodes were split by cell class(EXC, IN or GL). ChromA was run again for each cell class generating 3 sets of cell class specific peaks.

Aggregated chromatin accessibility profiles were used with TRANSFAC v2020.1 motifs and the inferelator-prior (v0.3.0) pipeline to create prior knowledge connectivity matrices between TFs and target genes for excitatory neurons, interneurons, and glial cells. Vascular cells were not present in the scATAC data sufficiently to allow construction of a vascular cell prior with this method, and so vascular cells were assigned the glial prior for network inference.

GRNs were learned using AMuSR on $\log_2$ pseudocount transformed count data for each of 36 cell type specific clusters as separate tasks with the appropriate prior knowledge network. An aggregate network was created by rank-summing each cell type GRN. MCC was calculated for this aggregate network based on a comparison to the union of the three prior knowledge networks, and the confidence score which maximized MCC was selected as a threshold to determine the size of the final network. Neuron specific edges were computed by aggregating filtered individual task networks with their respective confidence score to maximize MCC. Each edge that was shared with a glial or vascular network was removed. The remaining neuron specific edges are interneuron specific, excitatory specific or shared.

5.11 Inferelator 3.0 Single-Cell Computational Speed Profiling

144,682 mouse cells from the mouse neuronal subcluster EXC_IT_1 were used with the mouse excitatory neuron prior knowledge network to determine Inferelator 3.0 runtime. To benchmark the Dask engine, the Inferelator was deployed to 5 28-core (Intel® Xeon® E5-2690) nodes for a total of 140 cpu cores. To benchmark the python-based multiprocessing engine, the Inferelator was deployed to a single 28-core (Intel® Xeon® E5-2690) node. Either all 144,682 mouse cells were used, or a

subset was randomly selected for each run, and used to learn a single GRN. Runtime was determined by the length of workflow execution, which includes loading data, running all regressions, and producing output files.

5.12 Visualization

Figures were generated with R [89] and the common ggplot2 [90], umap [91], and tidyverse packages [92]. Additional figures were generated with python using scanpy [29], matplotlib [93], and seaborn [94]. Network diagrams were created with the python package jp_gene_viz [95]. Schematic figures were created in Adobe Illustrator, and other figures were adjusted in Illustrator to improve panelling and layout.

6 Declarations

Ethics Approval and Consent to Participate
Not applicable

Competing interests
The authors declare that they have no competing interests.

Availability of Data and Materials
The datasets supporting the conclusions of this article are available in the NCBI GEO repository with accession IDs: GSE125162, GSE144820, GSE67023, GSE27219, GSE142864. A large number of GEO records were compiled and normalized in a previous work [40] into a combined dataset which is available on Zenodo (DOI: 10.5281/zenodo.3247754). Single-cell mouse datasets are publicly available from 10x genomics [81, 84, 85, 86] under a Creative Commons Attribution (CC-BY 4.0) license. Software packages developed for this article are available on github (`https://github.com/flatironinstitute/inferelator` and `https://github.com/flatironinstitute/inferelator-prior`) and have been released as python packages through PyPi (`https://pypi.org/project/inferelator/` and `https://pypi.org/project/inferelator-prior/`). Specific analysis scripts for this work have been included in Supplemental Data 1.

Author's contributions
CSG contributed to Methodology, Software, Validation, Formal Analysis, Writing – Original Draft Preparation, and Visualization. CJ and GS contributed to Conceptualization, Methodology, Software, Validation, Investigation, Resources, Data Curation, Formal Analysis, Writing – Original Draft Preparation, and Visualization. AS contributed to Validation, Data Curation, Formal Analysis, and Visualization. AW contributed to Software and Visualization. AT contributed to Software, Writing – Original Draft Preparation, and Formal Analysis. DC and KT contributed to Software, Data Curation, and Conceptualization. NDV, NC, RY, and TH contributed to Software. DG contributed to Supervision, Project Administration, and Funding Acquisition. EM contributed to Conceptualization, Writing – Original Draft Preparation, and Software. RB contributed to Conceptualization, Writing – Original Draft Preparation, Supervision, Project Administration, and Funding Acquisition.

Funding
RB thanks the following sources for research support: NSF IOS-1546218, NIH R35GM122515, NIH R01HD096770, NIH R01NS116350, and the Simons Foundation. DG thanks the following sources for research support: NSF MCB-1818234, NIH R01GM107466, NIH R01GM134066, and NIH R01AI140766.

Acknowledgements
We thank past and present members of the Gresham, Miraldi, and Bonneau labs for discussions and valuable feedback on this manuscript. We also thank the staff of the Flatiron Institute Scientific Computing Core for their tireless efforts to build and maintain the High Performance Computing resources which we rely on. This work was supported in part through the NYU IT High Performance Computing resources, services, and staff expertise.

Additional Files
Supplemental Data 1 is a .tar.gz file containing the prior knowledge networks used in this work, the gold standard networks used in this work, and the python scripts used to generate the learned networks in this work
Supplemental Data 2 is a .tar.gz file containing the mouse E18 neuronal network learned in Figure 6 of this work
Supplemental Table 1 is a .tsv file containing the crossvalidation performance results from Figure 2 of this work
Supplemental Table 2 is a .tsv file containing the crossvalidation performance results from Figure 3 of this work
Supplemental Table 3 is a .tsv file containing the crossvalidation performance results from Figure 4 of this work

Author details
[1] Flatiron Institute, Center for Computational Biology, Simons Foundation, 10010, New York, USA. [2] Center For Genomics and Systems Biology, NYU, 10008, New York, USA. [3] Department of Biology, NYU, 10008, New York, USA. [4] Courant Institute of Mathematical Sciences, Computer Science Department, NYU, 10008, New York, USA. [5] Center For Data Science, NYU, 10008, New York, USA. [6] Department of Systems Biology, Columbia University, 10032, New York USA,. [7] Flatiron Institute, Scientific Computing Core, Simons Foundation, 10010, New York, USA. [8] Divisions of Immunobiology and Biomedical Informatics, Cincinnati Children's Hospital Medical Center, 45229, Cincinnati USA,. [9] Department of Pediatrics, University of Cincinnati College of Medicine, 45229, Cincinnati USA,.

References

1. Zaret, K.S.: Pioneer transcription factors initiating gene network changes. Annu. Rev. Genet. (2020)
2. Kadonaga, J.T.: Regulation of RNA polymerase II transcription by sequence-specific DNA binding factors. Cell **116**(2), 247–257 (2004)
3. Hahn, S., Young, E.T.: Transcriptional regulation in saccharomyces cerevisiae: transcription factor regulation and function, mechanisms of initiation, and roles of activators and coactivators. Genetics **189**(3), 705–736 (2011)
4. Lambert, S.A., Jolma, A., Campitelli, L.F., Das, P.K., Yin, Y., Albu, M., Chen, X., Taipale, J., Hughes, T.R., Weirauch, M.T.: The human transcription factors. Cell **172**(4), 650–665 (2018)
5. Thompson, D., Regev, A., Roy, S.: Comparative analysis of gene regulatory networks: from network reconstruction to evolution. Annu. Rev. Cell Dev. Biol. **31**, 399–428 (2015)
6. Chasman, D., Fotuhi Siahpirani, A., Roy, S.: Network-based approaches for analysis of complex biological systems. Curr. Opin. Biotechnol. **39**, 157–166 (2016)
7. Hu, J.X., Thomas, C.E., Brunak, S.: Network biology concepts in complex disease comorbidities. Nat. Rev. Genet. **17**(10), 615–629 (2016)
8. Mehta, T.K., Koch, C., Nash, W., Knaack, S.A., Sudhakar, P., Olbei, M., Bastkowski, S., Penso-Dolfin, L., Korcsmaros, T., Haerty, W., Roy, S., Di-Palma, F.: Evolution of regulatory networks associated with traits under selection in cichlids. Genome Biol. **22**(1), 25 (2021)
9. Peter, I.S., Davidson, E.H.: Evolution of gene regulatory networks controlling body plan development. Cell **144**(6), 970–985 (2011)
10. Huang, M., Bao, J., Hallström, B.M., Petranovic, D., Nielsen, J.: Efficient protein production by yeast requires global tuning of metabolism. Nat. Commun. **8**(1), 1131 (2017)
11. DeRisi, J.L., Iyer, V.R., Brown, P.O.: Exploring the metabolic and genetic control of gene expression on a genomic scale. Science **278**(5338), 680–686 (1997)
12. Nagalakshmi, U., Wang, Z., Waern, K., Shou, C., Raha, D., Gerstein, M., Snyder, M.: The transcriptional landscape of the yeast genome defined by RNA sequencing. Science **320**(5881), 1344–1349 (2008)
13. Wang, Z., Gerstein, M., Snyder, M.: RNA-Seq: a revolutionary tool for transcriptomics. Nat. Rev. Genet. **10**(1), 57–63 (2009)
14. Szederkényi, G., Banga, J.R., Alonso, A.A.: Inference of complex biological networks: distinguishability issues and optimization-based solutions. BMC Syst. Biol. **5**, 177 (2011)
15. Ud-Dean, S.M.M., Gunawan, R.: Optimal design of gene knockout experiments for gene regulatory network inference. Bioinformatics **32**(6), 875–883 (2016)
16. Hackett, S.R., Baltz, E.A., Coram, M., Wranik, B.J., Kim, G., Baker, A., Fan, M., Hendrickson, D.G., Berndl, M., McIsaac, R.S.: Learning causal networks using inducible transcription factors and transcriptome-wide time series. Mol. Syst. Biol. **16**(3), 9174 (2020)
17. Macosko, E.Z., Basu, A., Satija, R., Nemesh, J., Shekhar, K., Goldman, M., Tirosh, I., Bialas, A.R., Kamitaki, N., Martersteck, E.M., Trombetta, J.J., Weitz, D.A., Sanes, J.R., Shalek, A.K., Regev, A., McCarroll, S.A.: Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. Cell **161**(5), 1202–1214 (2015)
18. Zilionis, R., Nainys, J., Veres, A., Savova, V., Zemmour, D., Klein, A.M., Mazutis, L.: Single-cell barcoding and sequencing using droplet microfluidics. Nat. Protoc. **12**(1), 44 (2017)
19. Zheng, G.X.Y., Terry, J.M., Belgrader, P., Ryvkin, P., Bent, Z.W., Wilson, R., Ziraldo, S.B., Wheeler, T.D., McDermott, G.P., Zhu, J., Gregory, M.T., Shuga, J., Montesclaros, L., Underwood, J.G., Masquelier, D.A., Nishimura, S.Y., Schnall-Levin, M., Wyatt, P.W., Hindson, C.M., Bharadwaj, R., Wong, A., Ness, K.D., Beppu, L.W., Deeg, H.J., McFarland, C., Loeb, K.R., Valente, W.J., Ericson, N.G., Stevens, E.A., Radich, J.P., Mikkelsen, T.S., Hindson, B.J., Bielas, J.H.: Massively parallel digital transcriptional profiling of single cells. Nat. Commun. **8**, 14049 (2017)
20. Rosenberg, A.B., Roco, C.M., Muscat, R.A., Kuchina, A., Sample, P., Yao, Z., Graybuck, L.T., Peeler, D.J., Mukherjee, S., Chen, W., Pun, S.H., Sellers, D.L., Tasic, B., Seelig, G.: Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding. Science **360**(6385), 176–182 (2018)
21. Dixit, A., Parnas, O., Li, B., Chen, J., Fulco, C.P., Jerby-Arnon, L., Marjanovic, N.D., Dionne, D., Burks, T., Raychowdhury, R., Adamson, B., Norman, T.M., Lander, E.S., Weissman, J.S., Friedman, N., Regev, A.: Perturb-Seq: Dissecting molecular circuits with scalable Single-Cell RNA profiling of pooled genetic screens. Cell **167**(7), 1853–186617 (2016)
22. Jackson, C.A., Castro, D.M., Saldi, G.-A., Bonneau, R., Gresham, D.: Gene regulatory network reconstruction using single-cell RNA sequencing of barcoded genotypes in diverse environments. Elife **9**, 51254 (2020)
23. Schraivogel, D., Gschwind, A.R., Milbank, J.H., Leonce, D.R., Jakob, P., Mathur, L., Korbel, J.O., Merten, C.A., Velten, L., Steinmetz, L.M.: Targeted perturb-seq enables genome-scale genetic screens in single cells. Nat. Methods **17**(6), 629–635 (2020)
24. Ursu, O., Neal, J.T., Shea, E., Thakore, P.I., Jerby-Arnon, L., Nguyen, L., Dionne, D., Diaz, C., Bauman, J., Mosaad, M., Fagre, C., Giacomelli, A., Ly, S.H., Rozenblatt-Rosen, O., Hahn, W., Aguirre, A., Berger, A., Regev, A., Boehm, J.S.: Massively parallel phenotyping of variant impact in cancer with Perturb-seq reveals a shift in the spectrum of cell states induced by somatic mutations (2020)
25. Svensson, V.: Droplet scRNA-seq is not zero-inflated. Nat. Biotechnol. (2020)
26. Arisdakessian, C., Poirion, O., Yunits, B., Zhu, X., Garmire, L.X.: DeepImpute: an accurate, fast, and scalable deep neural network method to impute single-cell RNA-seq data. Genome Biol. **20**(1), 211 (2019)
27. Tjärnberg, A., Mahmood, O., Jackson, C.A., Saldi, G.-A., Cho, K., Christiaen, L.A., Bonneau, R.A.: Optimal tuning of weighted kNN- and diffusion-based methods for denoising single cell genomics data. PLoS Comput. Biol. **17**(1), 1008569 (2021)
28. Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck, W.M. 3rd, Hao, Y., Stoeckius, M., Smibert, P., Satija, R.: Comprehensive integration of Single-Cell data. Cell **177**(7), 1888–190221 (2019)

29. Wolf, F.A., Angerer, P., Theis, F.J.: SCANPY: large-scale single-cell gene expression data analysis. Genome Biol. **19**(1), 15 (2018)
30. Van de Sande, B., Flerin, C., Davie, K., De Waegeneer, M., Hulselmans, G., Aibar, S., Seurinck, R., Saelens, W., Cannoodt, R., Rouchon, Q., Verbeiren, T., De Maeyer, D., Reumers, J., Saeys, Y., Aerts, S.: A scalable SCENIC workflow for single-cell gene regulatory network analysis. Nat. Protoc. (2020)
31. Huynh-Thu, V.A., Irrthum, A., Wehenkel, L., Geurts, P.: Inferring regulatory networks from expression data using tree-based methods. PLoS One **5**(9) (2010)
32. Kamimoto, K., Hoffmann, C.M., Morris, S.A.: CellOracle: Dissecting cell identity via network inference and in silico gene perturbation (2020)
33. Matsumoto, H., Kiryu, H., Furusawa, C., Ko, M.S.H., Ko, S.B.H., Gouda, N., Hayashi, T., Nikaido, I.: SCODE: an efficient regulatory network inference algorithm from single-cell RNA-Seq during differentiation. Bioinformatics **33**(15), 2314–2321 (2017)
34. Bonneau, R., Reiss, D.J., Shannon, P., Facciotti, M., Hood, L., Baliga, N.S., Thorsson, V.: The inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. Genome Biol. **7**, 36 (2006)
35. Madar, A., Greenfield, A., Ostrer, H., Vanden-Eijnden, E., Bonneau, R.: The inferelator 2.0: A scalable framework for reconstruction of dynamic regulatory network models. In: 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 5448–5451 (2009)
36. Arrieta-Ortiz, M.L., Hafemeister, C., Bate, A.R., Chu, T., Greenfield, A., Shuster, B., Barry, S.N., Gallitto, M., Liu, B., Kacmarczyk, T., Santoriello, F., Chen, J., Rodrigues, C.D.A., Sato, T., Rudner, D.Z., Driks, A., Bonneau, R., Eichenberger, P.: An experimentally supported model of the bacillus subtilis global transcriptional regulatory network. Mol. Syst. Biol. **11**(11), 839 (2015)
37. Ciofani, M., Madar, A., Galan, C., Sellars, M., Mace, K., Pauli, F., Agarwal, A., Huang, W., Parkurst, C.N., Muratet, M., Newberry, K.M., Meadows, S., Greenfield, A., Yang, Y., Jain, P., Kirigin, F.K., Birchmeier, C., Wagner, E.F., Murphy, K.M., Myers, R.M., Bonneau, R., Littman, D.R.: A validated regulatory network for th17 cell specification. Cell **151**(2), 289–303 (2012)
38. Miraldi, E.R., Pokrovskii, M., Watters, A., Castro, D.M., De Veaux, N., Hall, J.A., Lee, J.-Y., Ciofani, M., Madar, A., Carriero, N., Littman, D.R., Bonneau, R.: Leveraging chromatin accessibility for transcriptional regulatory network inference in T helper 17 cells. Genome Res. (2019)
39. Pokrovskii, M., Hall, J.A., Ochayon, D.E., Yi, R., Chaimowitz, N.S., Seelamneni, H., Carriero, N., Watters, A., Waggoner, S.N., Littman, D.R., Bonneau, R., Miraldi, E.R.: Characterization of transcriptional regulatory networks that promote and restrict identities and functions of intestinal innate lymphoid cells. Immunity **51**(1), 185–1976 (2019)
40. Tchourine, K., Vogel, C., Bonneau, R.: Condition-Specific modeling of biophysical parameters advances inference of regulatory networks. Cell Rep. **23**(2), 376–388 (2018)
41. Wilkins, O., Hafemeister, C., Plessis, A., Holloway-Phillips, M.-M., Pham, G.M., Nicotra, A.B., Gregorio, G.B., Jagadish, S.V.K., Septiningsih, E.M., Bonneau, R., Purugganan, M.: EGRINs (environmental gene regulatory influence networks) in rice that function in the response to water deficit, high temperature, and agricultural environments. Plant Cell **28**(10), 2365–2384 (2016)
42. Rocklin, M.: Dask: Parallel computation with blocked algorithms and task scheduling. In: Proceedings of the 14th Python in Science Conference. Proceedings of the Python in Science Conference, pp. 126–132. SciPy, ??? (2015)
43. Nicolas, P., Mäder, U., Dervyn, E., Rochat, T., Leduc, A., Pigeonneau, N., Bidnenko, E., Marchadier, E., Hoebeke, M., Aymerich, S., Becher, D., Bisicchia, P., Botella, E., Delumeau, O., Doherty, G., Denham, E.L., Fogg, M.J., Fromion, V., Goelzer, A., Hansen, A., Härtig, E., Harwood, C.R., Homuth, G., Jarmer, H., Jules, M., Klipp, E., Le Chat, L., Lecointe, F., Lewis, P., Liebermeister, W., March, A., Mars, R.A.T., Nannapaneni, P., Noone, D., Pohl, S., Rinn, B., Rügheimer, F., Sappa, P.K., Samson, F., Schaffer, M., Schwikowski, B., Steil, L., Stülke, J., Wiegert, T., Devine, K.M., Wilkinson, A.J., van Dijl, J.M., Hecker, M., Völker, U., Bessières, P., Noirot, P.: Condition-dependent transcriptome reveals high-level regulatory architecture in bacillus subtilis. Science **335**(6072), 1103–1106 (2012)
44. Greenfield, A., Madar, A., Ostrer, H., Bonneau, R.: DREAM4: Combining genetic and dynamic information to identify biological networks and dynamical models. PLoS One **5**(10), 13397 (2010)
45. Castro, D.M., de Veaux, N.R., Miraldi, E.R., Bonneau, R.: Multi-study inference of regulatory networks for more accurate models of gene regulation. PLoS Comput. Biol. **15**(1), 1006591 (2019)
46. ENCODE Project Consortium, Moore, J.E., Purcaro, M.J., Pratt, H.E., Epstein, C.B., Shoresh, N., Adrian, J., Kawli, T., Davis, C.A., Dobin, A., Kaul, R., Halow, J., Van Nostrand, E.L., Freese, P., Gorkin, D.U., Shen, Y., He, Y., Mackiewicz, M., Pauli-Behn, F., Williams, B.A., Mortazavi, A., Keller, C.A., Zhang, X.-O., Elhajjajy, S.I., Huey, J., Dickel, D.E., Snetkova, V., Wei, X., Wang, X., Rivera-Mulia, J.C., Rozowsky, J., Zhang, J., Chhetri, S.B., Zhang, J., Victorsen, A., White, K.P., Visel, A., Yeo, G.W., Burge, C.B., Lécuyer, E., Gilbert, D.M., Dekker, J., Rinn, J., Mendenhall, E.M., Ecker, J.R., Kellis, M., Klein, R.J., Noble, W.S., Kundaje, A., Guigó, R., Farnham, P.J., Cherry, J.M., Myers, R.M., Ren, B., Graveley, B.R., Gerstein, M.B., Pennacchio, L.A., Snyder, M.P., Bernstein, B.E., Wold, B., Hardison, R.C., Gingeras, T.R., Stamatoyannopoulos, J.A., Weng, Z.: Expanded encyclopaedias of DNA elements in the human and mouse genomes. Nature **583**(7818), 699–710 (2020)
47. Jariani, A., Vermeersch, L., Cerulus, B., Perez-Samper, G., Voordeckers, K., Van Brussel, T., Thienpont, B., Lambrechts, D., Verstrepen, K.J.: A new protocol for single-cell RNA-seq reveals stochastic gene expression during lag phase in budding yeast. Elife **9** (2020)
48. Sun, Z., Xu, X., He, J., Murray, A., Sun, M.-a., Wei, X., Wang, X., McCoig, E., Xie, E., Jiang, X., Li, L., Zhu, J., Chen, J., Morozov, A., Pickrell, A.M., Theus, M.H., Xie, H.: Egr1 recruits tet1 to shape the brain methylome during development and upon neuronal activity. Nature Communications **10**(1), 3892 (2019). doi:10.1038/s41467-019-11905-3
49. Liu, J., Pasini, S., Shelanski, M.L., Greene, L.A.: Activating transcription factor 4 (ATF4) modulates

post-synaptic development and dendritic spine morphology. Front. Cell. Neurosci. **8**, 177 (2014)
50. Saito, Y., Miranda-Rottmann, S., Ruggiu, M., Park, C.Y., Fak, J.J., Zhong, R., Duncan, J.S., Fabella, B.A., Junge, H.J., Chen, Z., Araya, R., Fritzsch, B., Hudspeth, A.J., Darnell, R.B.: NOVA2-mediated RNA regulation is required for axonal pathfinding during development. Elife **5** (2016)
51. Wortel, I.M.N., van der Meer, L.T., Kilberg, M.S., van Leeuwen, F.N.: Surviving stress: Modulation of atf4-mediated stress responses in normal and malignant cells. Trends in Endocrinology & Metabolism **28**(11), 794–806 (2017). doi:10.1016/j.tem.2017.07.003
52. Corona, C., Pasini, S., Liu, J., Amar, F., Greene, L.A., Shelanski, M.L.: Activating transcription factor 4 (atf4) regulates neuronal activity by controlling gababr trafficking. Journal of Neuroscience **38**(27), 6102–6113 (2018). doi:10.1523/JNEUROSCI.3350-17.2018. https://www.jneurosci.org/content/38/27/6102.full.pdf
53. Oh, C.-K., Choi, Y.K., Hwang, I.-Y., Ko, Y.U., Chung, I.K., Yun, N., Oh, Y.J.: RING-finger protein 166 plays a novel pro-apoptotic role in neurotoxin-induced neurodegeneration via ubiquitination of XIAP. Cell Death Dis. **11**(10), 939 (2020)
54. Culotta, L., Scalmani, P., Vinci, E., Terragni, B., Sessa, A., Broccoli, V., Mantegazza, M., Boeckers, T., Verpelli, C.: Sult4a1 modulates synaptic development and function by promoting the formation of psd-95/nmdar complex. Journal of Neuroscience **40**(37), 7013–7026 (2020). doi:10.1523/JNEUROSCI.2194-19.2020. https://www.jneurosci.org/content/40/37/7013.full.pdf
55. Papatsenko, D., Levine, M.S.: Dual regulation by the hunchback gradient in the drosophila embryo. Proc. Natl. Acad. Sci. U. S. A. **105**(8), 2901–2906 (2008)
56. Lam, K.Y., Westrick, Z.M., Müller, C.L., Christiaen, L., Bonneau, R.: Fused regression for multi-source gene regulatory network inference. PLoS Comput. Biol. **12**(12), 1005157 (2016)
57. 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.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., Del Río, J.F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E.: Array programming with NumPy. Nature **585**(7825), 357–362 (2020)
58. Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., Vijaykumar, A., Bardelli, A.P., Rothberg, A., Hilboll, A., Kloeckner, A., Scopatz, A., Lee, A., Rokem, A., Woods, C.N., Fulton, C., Masson, C., Häggström, C., Fitzgerald, C., Nicholson, D.A., Hagen, D.R., Pasechnik, D.V., Olivetti, E., Martin, E., Wieser, E., Silva, F., Lenders, F., Wilhelm, F., Young, G., Price, G.A., Ingold, G.-L., Allen, G.E., Lee, G.R., Audren, H., Probst, I., Dietrich, J.P., Silterra, J., Webber, J.T., Slavič, J., Nothman, J., Buchner, J., Kulick, J., Schönberger, J.L., de Miranda Cardoso, J.V., Reimer, J., Harrington, J., Rodríguez, J.L.C., Nunez-Iglesias, J., Kuczynski, J., Tritz, K., Thoma, M., Newville, M., Kümmerer, M., Bolingbroke, M., Tartre, M., Pak, M., Smith, N.J., Nowaczyk, N., Shebanov, N., Pavlyk, O., Brodtkorb, P.A., Lee, P., McGibbon, R.T., Feldbauer, R., Lewis, S., Tygier, S., Sievert, S., Vigna, S., Peterson, S., More, S., Pudlik, T., Oshima, T., Pingel, T.J., Robitaille, T.P., Spura, T., Jones, T.R., Cera, T., Leslie, T., Zito, T., Krauss, T., Upadhyay, U., Halchenko, Y.O., Vázquez-Baeza, Y., SciPy 1.0 Contributors: SciPy 1.0: fundamental algorithms for scientific computing in python. Nat. Methods (2020)
59. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É.: Scikit-learn: Machine learning in python. J. Mach. Learn. Res. **12**(Oct), 2825–2830 (2011)
60. Quinlan, A.R., Hall, I.M.: BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics **26**(6), 841–842 (2010)
61. Dale, R.K., Pedersen, B.S., Quinlan, A.R.: Pybedtools: a flexible python library for manipulating genomic datasets and annotations. Bioinformatics **27**(24), 3423–3424 (2011)
62. Lambert, S.A., Yang, A.W.H., Sasse, A., Cowley, G., Albu, M., Caddick, M.X., Morris, Q.D., Weirauch, M.T., Hughes, T.R.: Similarity regression predicts evolution of transcription factor sequence specificity. Nat. Genet. **51**(6), 981–989 (2019)
63. Fornes, O., Castro-Mondragon, J.A., Khan, A., van der Lee, R., Zhang, X., Richmond, P.A., Modi, B.P., Correard, S., Gheorghe, M., Baranašić, D., Santana-Garcia, W., Tan, G., Chèneby, J., Ballester, B., Parcy, F., Sandelin, A., Lenhard, B., Wasserman, W.W., Mathelier, A.: JASPAR 2020: update of the open-access database of transcription factor binding profiles. Nucleic Acids Res. **48**(D1), 87–92 (2020)
64. Matys, V., Kel-Margoulis, O.V., Fricke, E., Liebich, I., Land, S., Barre-Dirrie, A., Reuter, I., Chekmenev, D., Krull, M., Hornischer, K., Voss, N., Stegmaier, P., Lewicki-Potapov, B., Saxel, H., Kel, A.E., Wingender, E.: TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes. Nucleic Acids Res. **34**(Database issue), 108–10 (2006)
65. Teixeira, M.C., Monteiro, P.T., Palma, M., Costa, C., Godinho, C.P., Pais, P., Cavalheiro, M., Antunes, M., Lemos, A., Pedreira, T., Sá-Correia, I.: YEASTRACT: an upgraded database for the analysis of transcription regulatory networks in saccharomyces cerevisiae. Nucleic Acids Res. **46**(D1), 348–353 (2018)
66. Monteiro, P.T., Oliveira, J., Pais, P., Antunes, M., Palma, M., Cavalheiro, M., Galocha, M., Godinho, C.P., Martins, L.C., Bourbon, N., Mota, M.N., Ribeiro, R.A., Viana, R., Sá-Correia, I., Teixeira, M.C.: YEASTRACT+: a portal for cross-species comparative genomics of transcription regulation in yeasts. Nucleic Acids Res. **48**(D1), 642–649 (2020)
67. Grant, C.E., Bailey, T.L., Noble, W.S.: FIMO: scanning for occurrences of a given motif. Bioinformatics **27**(7), 1017–1018 (2011)
68. Bailey, T.L., Boden, M., Buske, F.A., Frith, M., Grant, C.E., Clementi, L., Ren, J., Li, W.W., Noble, W.S.: MEME SUITE: tools for motif discovery and searching. Nucleic Acids Res. **37**(Web Server issue), 202–8 (2009)
69. Kim, J.T., Martinetz, T., Polani, D.: Bioinformatic principles underlying the information content of transcription factor binding sites. J. Theor. Biol. **220**(4), 529–544 (2003)

70. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. KDD'96, pp. 226–231. AAAI Press, ??? (1996)
71. Fu, Y., Jarboe, L.R., Dickerson, J.A.: Reconstructing genome-wide regulatory network of e. coli using transcriptome data and predicted transcription factor activities. BMC Bioinformatics **12**, 233 (2011)
72. Madar, A., Greenfield, A., Vanden-Eijnden, E., Bonneau, R.: DREAM3: Network inference using dynamic context likelihood of relatedness and the inferelator. PLoS One **5**(3), 9803 (2010)
73. Greenfield, A., Hafemeister, C., Bonneau, R.: Robust data-driven incorporation of prior knowledge into the inference of dynamic regulatory networks. Bioinformatics **29**(8), 1060–1067 (2013)
74. Wes McKinney: Data Structures for Statistical Computing in Python. In: Stéfan van der Walt, Jarrod Millman (eds.) Proceedings of the 9th Python in Science Conference, pp. 56–61 (2010). doi:10.25080/Majora-92bf1922-00a
75. Schacht, T., Oswald, M., Eils, R., Eichmüller, S.B., König, R.: Estimating the activity of transcription factors by the effect on their target genes. Bioinformatics **30**(17), 401–7 (2014)
76. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. J. R. Stat. Soc. Series B Stat. Methodol. **67**(2), 301–320 (2005)
77. Zou, H.: The adaptive lasso and its oracle properties. J. Am. Stat. Assoc. **101**(476), 1418–1429 (2006)
78. Liu, H., Roeder, K., Wasserman, L.: Stability approach to regularization selection (StARS) for high dimensional graphical models (2010). 1006.3316
79. Matthews, B.W.: Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochim. Biophys. Acta **405**(2), 442–451 (1975)
80. Ritchie, M.E., Phipson, B., Wu, D., Hu, Y., Law, C.W., Shi, W., Smyth, G.K.: limma powers differential expression analyses for RNA-sequencing and microarray studies. Nucleic Acids Res. **43**(7), 47 (2015)
81. 10x Genomics: 1.3 Million Brain Cells from E18 Mice. `https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.3.0/1M_neurons` (2017)
82. Satija, R., Farrell, J.A., Gennert, D., Schier, A.F., Regev, A.: Spatial reconstruction of single-cell gene expression data. Nat. Biotechnol. **33**(5), 495–502 (2015)
83. Di Bella, D.J., Habibi, E., Yang, S.-M., Stickels, R.R., Brown, J., Yadollahpour, P., Chen, F., Macosko, E.Z., Regev, A., Arlotta, P.: Molecular Logic of Cellular Diversification in the Mammalian Cerebral Cortex (2020)
84. 10x Genomics: Fresh cortex, hippocampus, and ventricular zone from embryonic mouse brain (E18). `https://support.10xgenomics.com/single-cell-atac/datasets/1.2.0/atac_v1_E18_brain_fresh_5k` (2019)
85. 10x Genomics: Dissociated and cryopreserved cortex, hippocampus, and ventricular zone cells from embryonic mouse brain (E18). `https://support.10xgenomics.com/single-cell-atac/datasets/1.2.0/atac_v1_E18_brain_cryo_5k` (2019)
86. 10x Genomics: Flash frozen cortex, hippocampus, and ventricular zone from embryonic mouse brain (E18). `https://support.10xgenomics.com/single-cell-atac/datasets/1.2.0/atac_v1_E18_brain_flash_5k` (2019)
87. Gabitto, M.I., Rasmussen, A., Wapinski, O., Allaway, K., Carriero, N., Fishell, G.J., Bonneau, R.: Characterizing chromatin landscape from aggregate and single-cell genomic assays using flexible duration modeling. Nature Communications **11**(1), 747 (2020). doi:10.1038/s41467-020-14497-5
88. Fang, R., Preissl, S., Li, Y., Hou, X., Lucero, J., Wang, X., Motamedi, A., Shiau, A.K., Zhou, X., Xie, F., Mukamel, E.A., Zhang, K., Zhang, Y., Behrens, M.M., Ecker, J.R., Ren, B.: Comprehensive analysis of single cell atac-seq data with snapatac. Nature Communications **12**(1), 1337 (2021). doi:10.1038/s41467-021-21583-9
89. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2020). R Foundation for Statistical Computing. https://www.R-project.org/
90. Wickham, H.: Ggplot2: Elegant Graphics for Data Analysis. Springer, ??? (2016). https://ggplot2.tidyverse.org
91. McInnes, L., Healy, J., Melville, J.: UMAP: Uniform manifold approximation and projection for dimension reduction. arXiv:1802.03426 [cs, stat] (2018)
92. Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T., Miller, E., Bache, S., Müller, K., Ooms, J., Robinson, D., Seidel, D., Spinu, V., Takahashi, K., Vaughan, D., Wilke, C., Woo, K., Yutani, H.: Welcome to the tidyverse. J. Open Source Softw. **4**(43), 1686 (2019)
93. Hunter, J.D.: Matplotlib: A 2D graphics environment. Computing in Science Engineering **9**(3), 90–95 (2007)
94. Waskom, M.L.: seaborn: statistical data visualization. Journal of Open Source Software **6**(60), 3021 (2021). doi:10.21105/joss.03021
95. Watters, A.: jp_gene_viz. `https://github.com/simonsfoundation/jp\_gene\_viz` (2019)

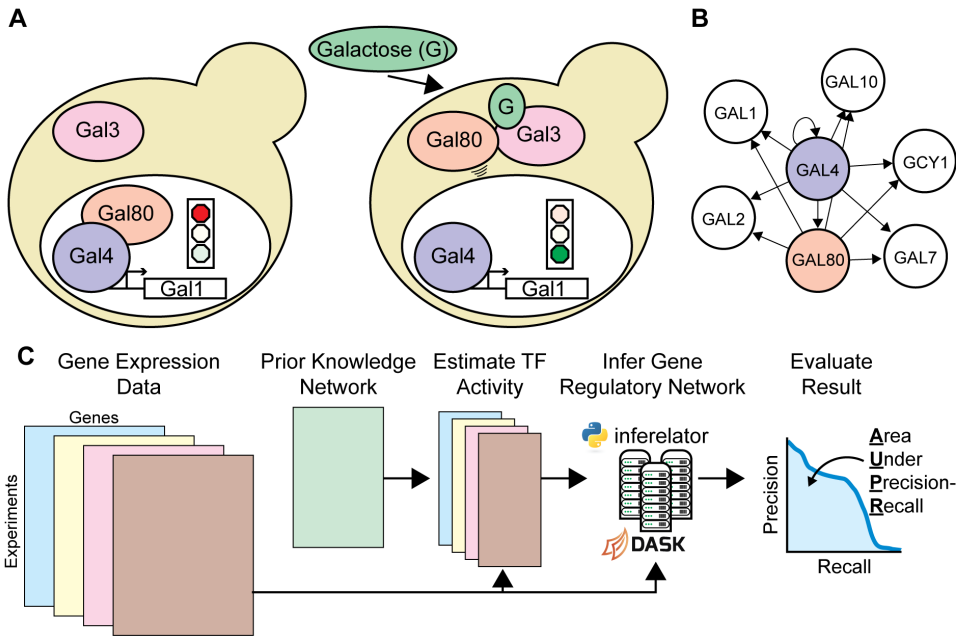


Figure 1: Learning Gene Regulatory Networks with the Inferelator (**A**) The response to the sugar galactose in *Saccharomyces cerevisiae* is mediated by the Gal4 and Gal80 TFs, a prototypical mechanism for altering cellular gene expression in response to stimuli. (**B**) Gal4 and Gal80 regulation represented as an unsigned directed graph connecting regulatory TFs to target genes. (**C**) Genome-wide Gene Regulatory Networks (GRNs) are inferred from gene expression data and prior knowledge about network connections using the Inferelator, and the resulting networks are scored by comparison with a gold standard of known interactions.

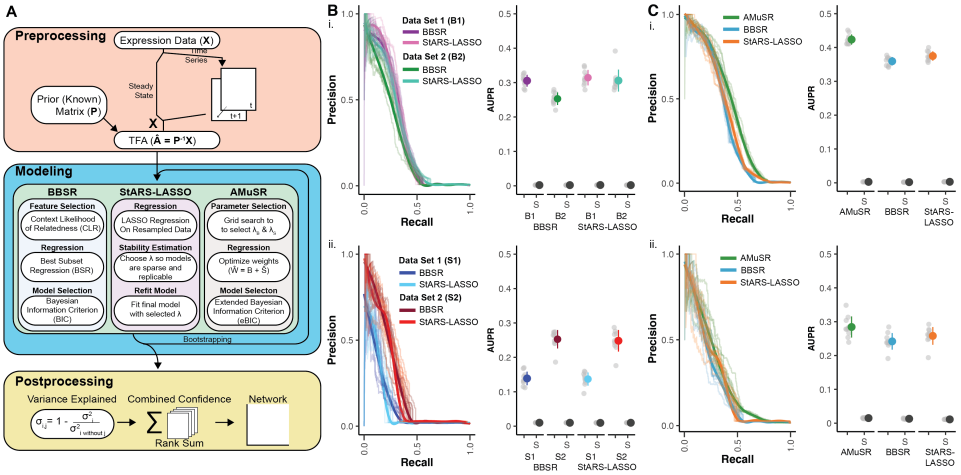


Figure 2: Network Inference Performance on Multiple Model Organism Datasets (**A**) Schematic of Inferelator workflow and a brief summary of the differences between GRN model selection methods (**B**) Results from 10 replicates of GRN inference for each modeling method on (i) *Bacillus subtilis* GSE67023 (B1), GSE27219 (B2) and (ii) *Saccharomyces cerevisiae* GSE142864 (S1), and [40] (S2). Precision-recall curves are shown for replicates where 20% of genes are held out of the prior and used for evaluation, with a smoothed consensus curve. AUPR is plotted for each cross-validation result in gray, with mean ± standard deviation in color. Experiments labeled with (S) are shuffled controls, where the labels on the prior adjacency matrix have been randomly shuffled. 10 shuffled replicates are shown as gray dots, with mean ± standard deviation in black. (**C**) Results from 10 replicates of GRN inference using two datasets as two network inference tasks on (i) *Bacillus subtilis* and (ii) *Saccharomyces cerevisiae*. AMuSR is a multi-task learning method; BBSR and StARS-LASSO are run on each task separately and then combined into a unified GRN. Precision-recall curves and AUPR are plotted as in **B**.

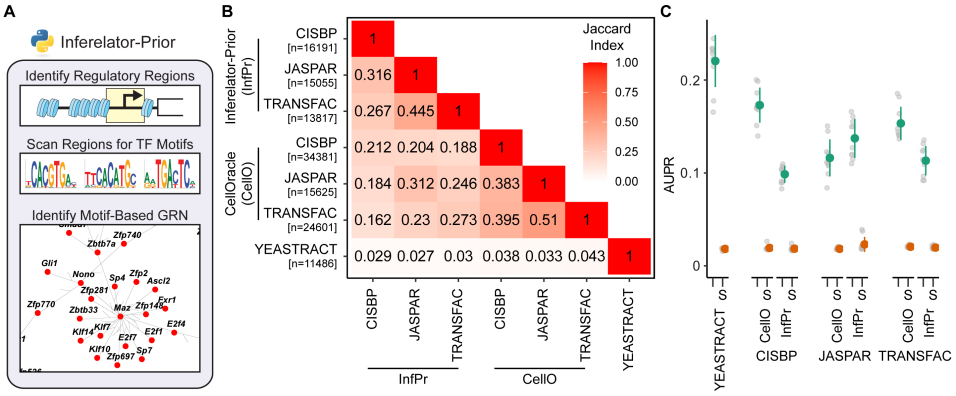


Figure 3: Construction and Performance of Network Connectivity Priors Using TF Motif Scanning (**A**) Schematic of inferelator-prior workflow, scanning identified regulatory regions (e.g. by ATAC) for TF motifs to construct adjacency matrices (**B**) Jaccard similarity index between *Saccharomyces cerevisiae* prior adjacency matrices generated by the inferelator-prior package, by the CellOracle package, and obtained from the YEASTRACT database. Prior matrices were generated using TF motifs from the CIS-BP, JASPAR, and TRANSFAC databases with each pipeline (n is the number of edges in each prior adjacency matrix). (**C**) The performance of Inferelator network inference using each motif-derived prior. Performance is evaluated by AUPR, scoring against genes held out of the prior adjacency matrix, based on inference using 2577 genome-wide microarray experiments. Experiments labeled with (S) are shuffled controls, where the labels on the prior adjacency matrix have been randomly shuffled.

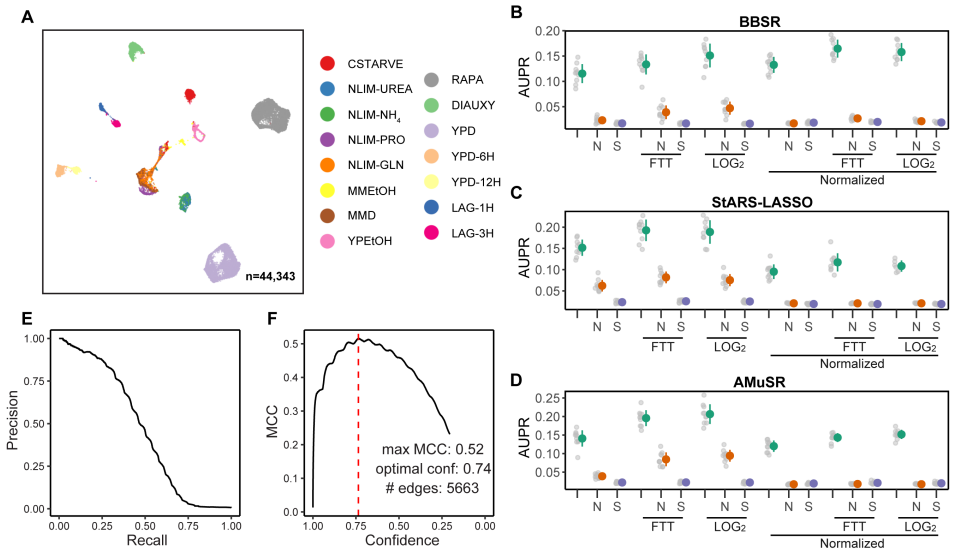


Figure 4: Network Inference Performance Using Single-Cell *Saccharomyces cerevisiae* Expression Data (**A**) Uniform Manifold Approximation and Projection (UMAP) plot of single-cell yeast data, colored by the experimental grouping of individual cells (tasks). (**B**) The effect of preprocessing methods on network inference using BBSR model selection on 14 task-specific expression datasets, as measured by AUPR. Colored dots represent mean ± standard deviation of all replicates. Data is either untransformed (raw counts), transformed by Freeman-Tukey Transform (FTT), or transformed by $log_2(x_1)$ pseudocount. Non-normalized data is compared to data normalized so that all cells have identical count depth. Network inference performance is compared to two baseline controls; data which has been replaced by Gaussian noise (N) and network inference using shuffled labels in the prior network (S). (**C**) Performance evaluated as in **B** on StARS-LASSO model selection. (**D**) Performance evaluated as in **B** on AMuSR model selection. (**E**) Precision-recall of the recovery of the prior on a final network constructed using FTT-transformed, non-normalized AMuSR model selection. (**F**) Matthews Correlation Coefficient (MCC) of the same network as in **E**

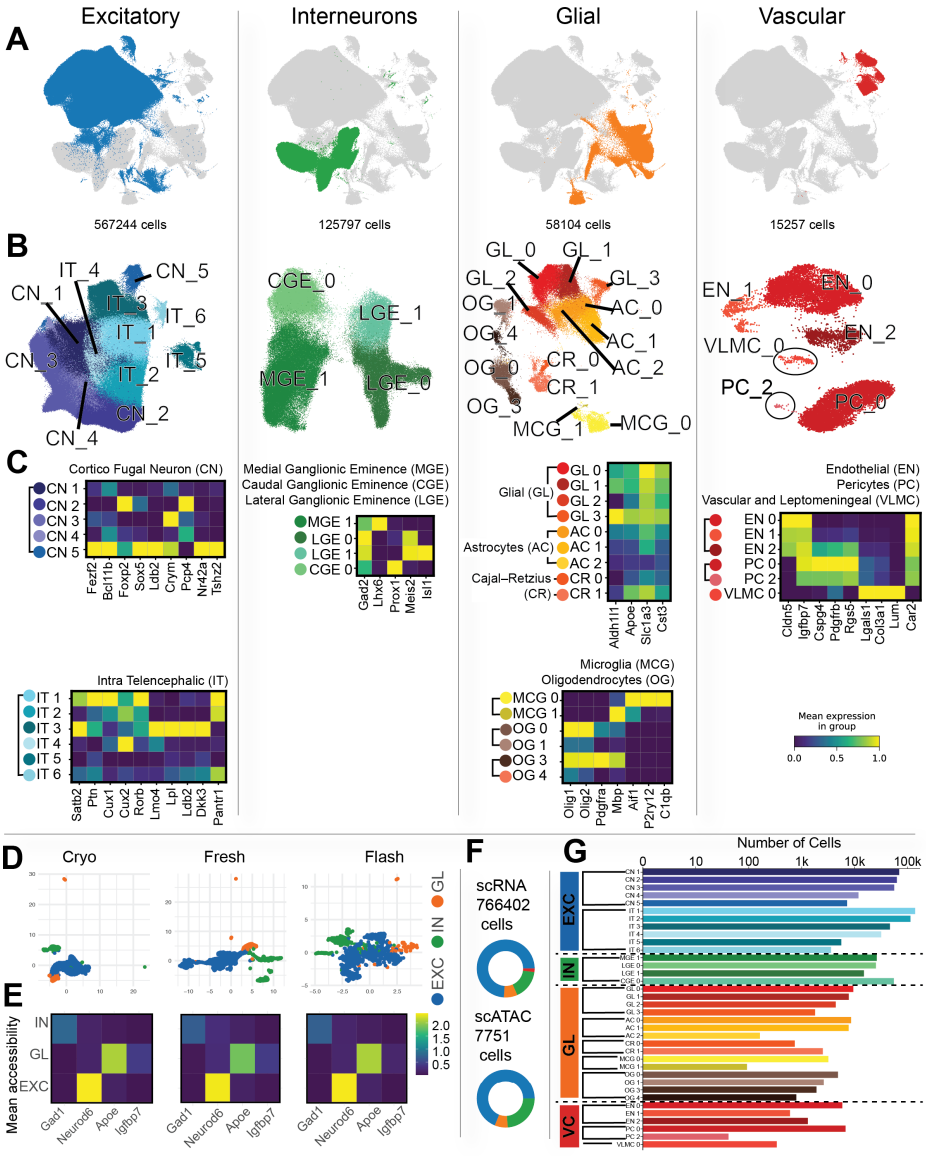


Figure 5: Processing Large Single-Cell Mouse Brain Data for Network Inference (**A**) UMAP plot of all mouse brain scRNA-seq data with Excitatory neurons, Interneurons, Glial cells and Vascular cells colored. (**B**) UMAP plot of cells from each broad category colored by louvain clusters and labeled by cell type. (**C**) Heatmap of normalized gene expression for marker genes that distinguish cluster cell types within broad categories. (**D**) UMAP plot of mouse brain scATAC data with Excitatory neurons, Interneurons, and Glial cells colored. (**E**) Heatmap of normalized mean gene accessibility for marker genes that distinguish broad categories of cells. (**F**) The number of scRNA-seq and scATAC cells in each of the broad categories. (**G**) The number of scRNA-seq cells in each cell type specific cluster.

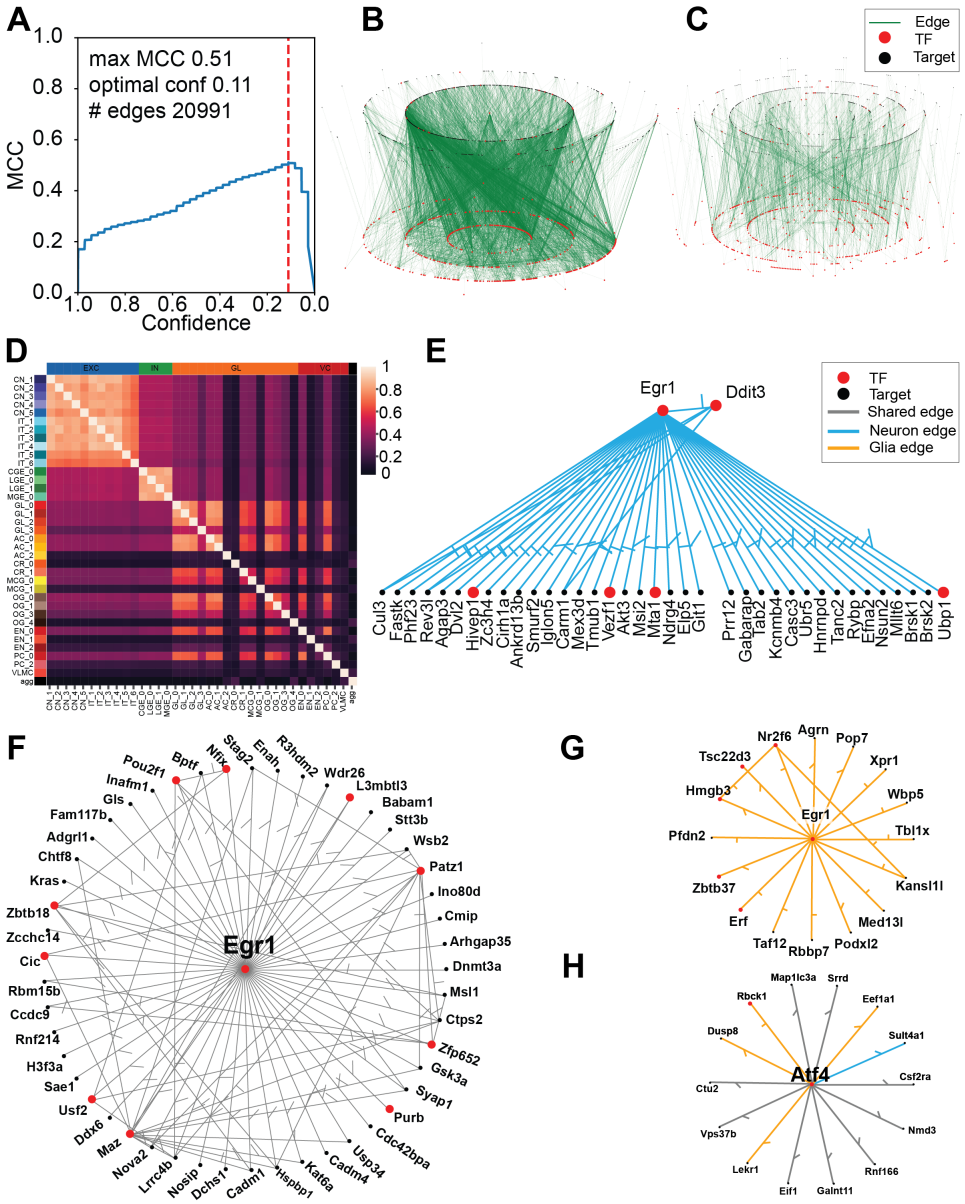
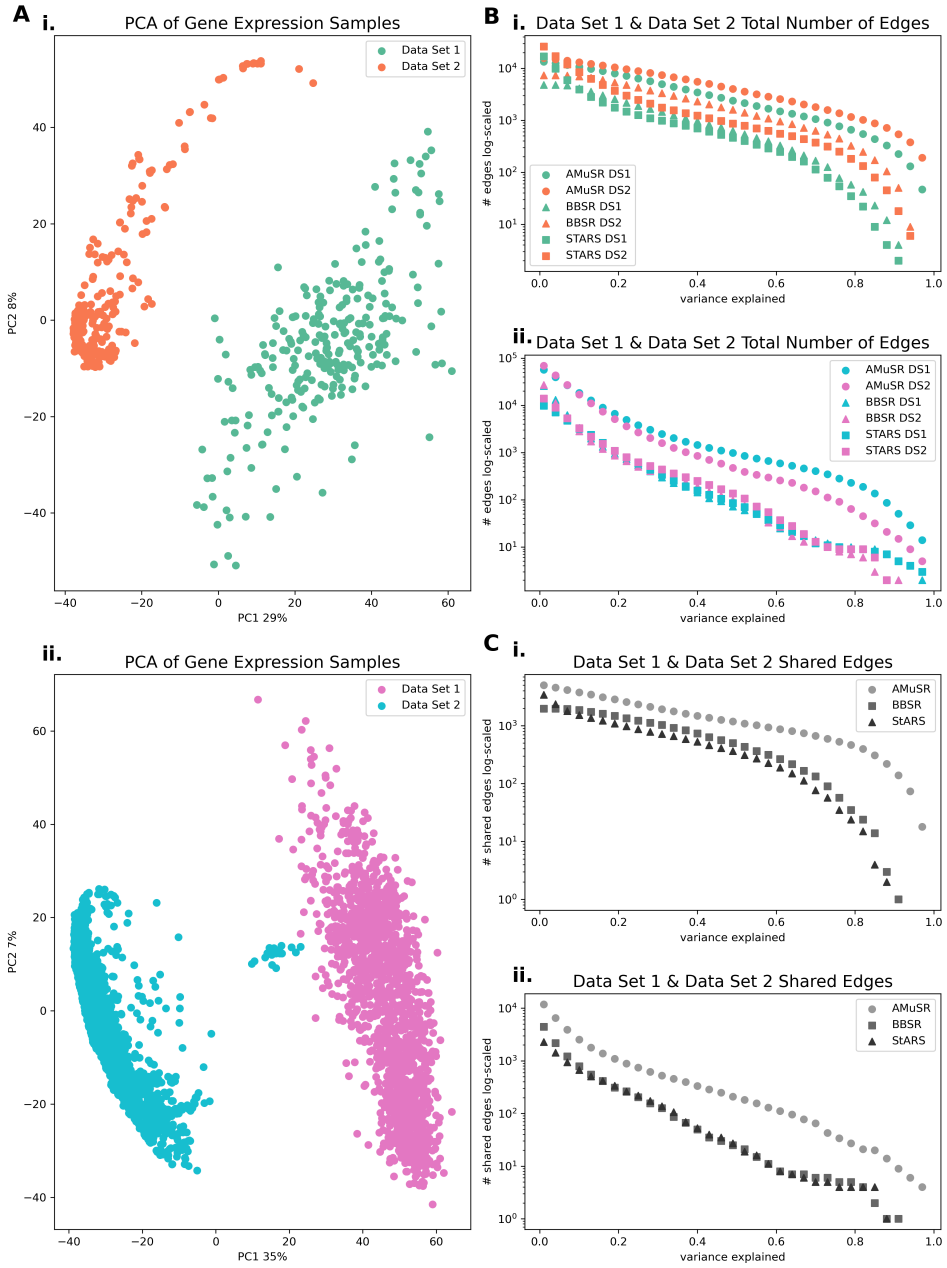
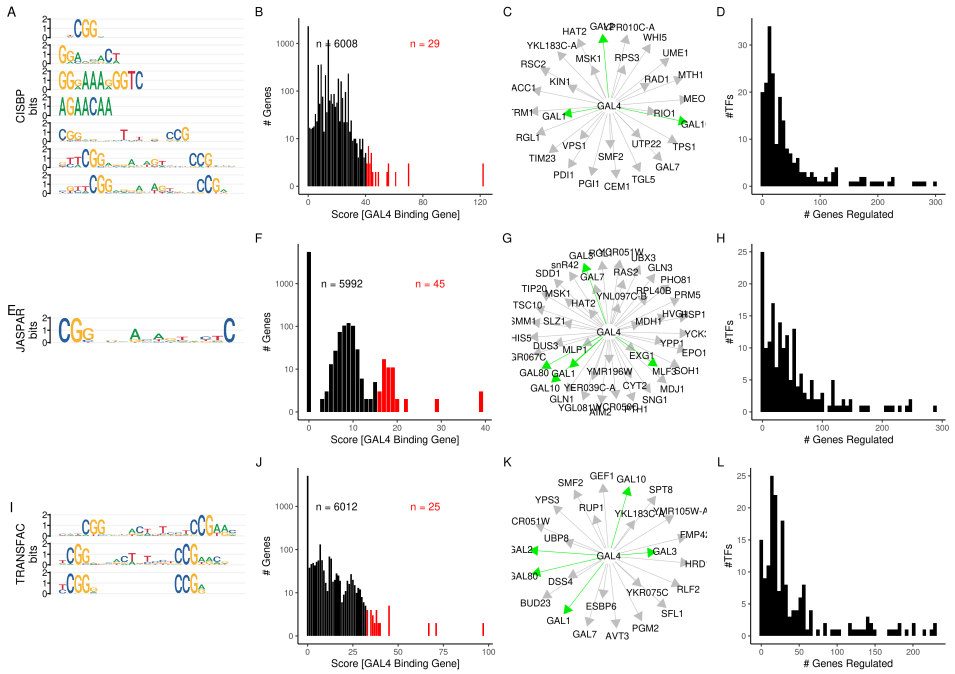


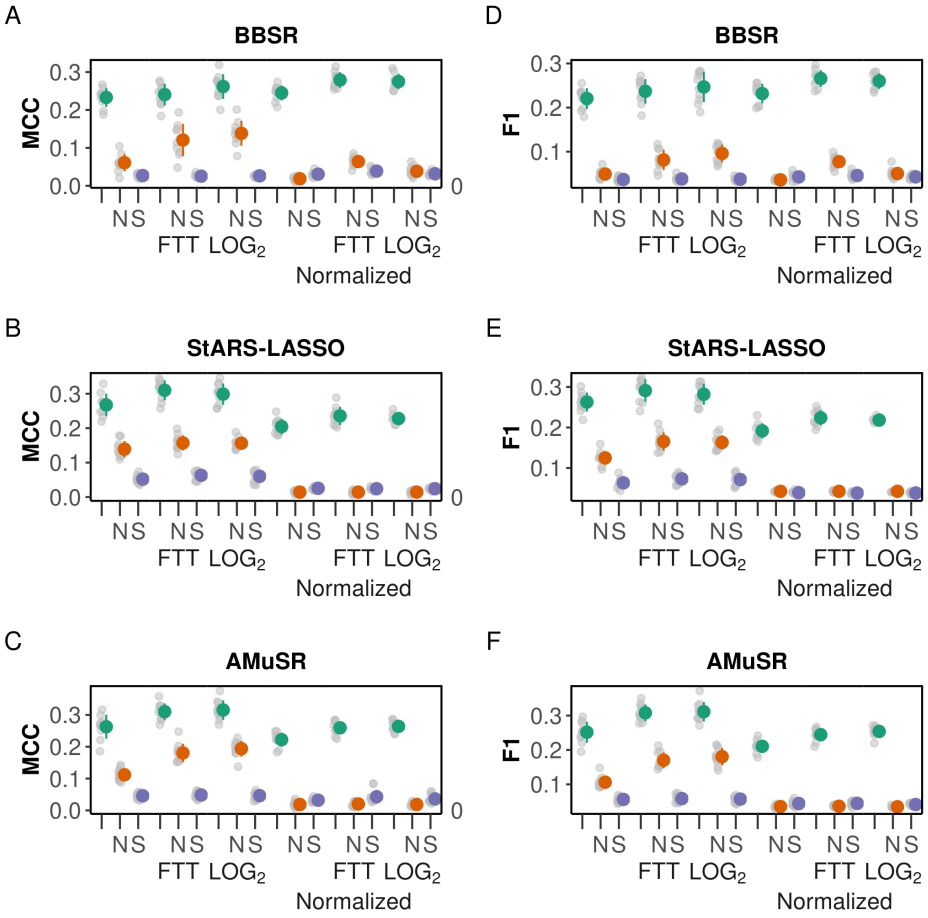
Figure 6: Learned GRN For The Mouse Brain (**A**) MCC for the aggregate network based on Inferelator prediction confidence. The dashed line shows the confidence score which maximizes MCC. Network edges at and above this line are retained in the final network. (**B**) Aggregate GRN learned. (**C**) Network edges which are present in every individual task. (**D**) Jaccard similarity index between each task network (**E**) Network targets of the *EGR1* TF in neurons. (**F**) Network targets of the *EGR1* TF in both neurons and glial cells. (**G**) Network targets of the *EGR1* TF in glial cells. (**H**) Network of the *ATF4* TF where blue edges are neuron specific, orange edges are glial specific, and black edges are present in both categories.



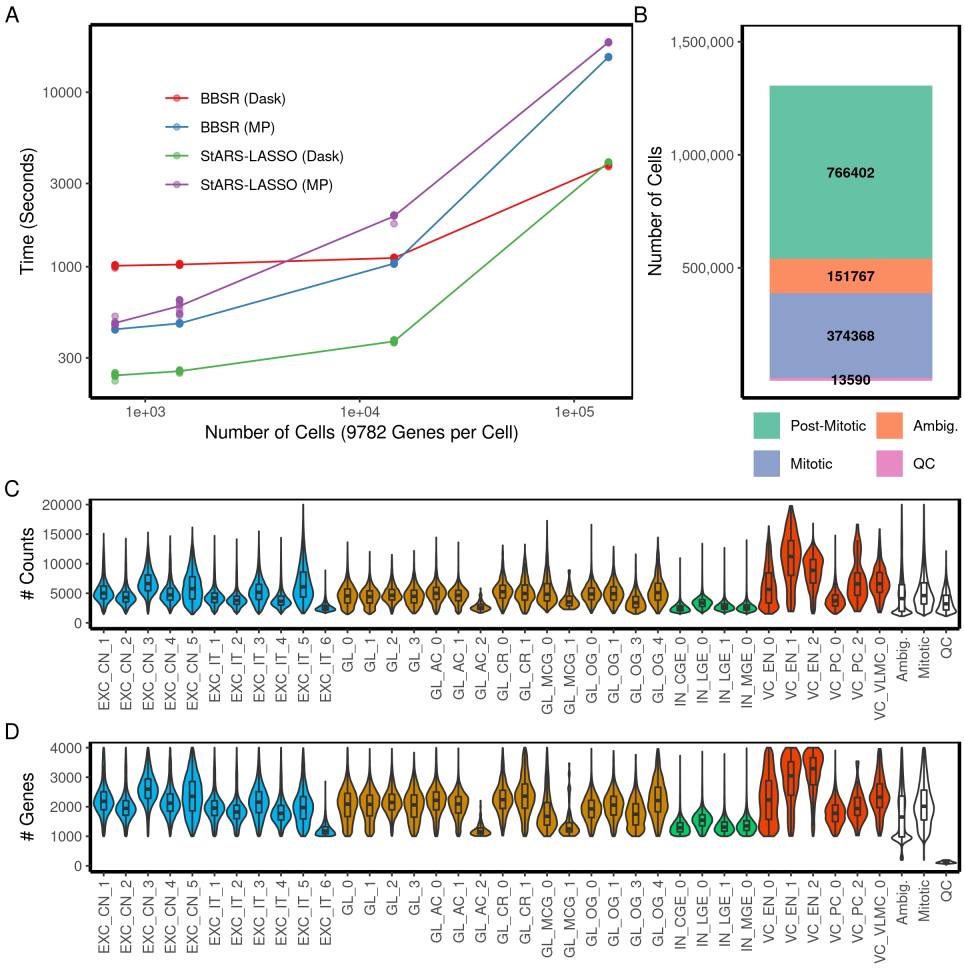
Supplemental Figure 1: Learning *Bacillus subtilis* and *Saccharomyces cerevisiae* networks by tasks. (A) PCA depicts batch effects between datasets for both (i) *Bacillus subtilis* and (ii) *Saccharomyces cerevisiae*. Learning networks by treating the independently collected datasets as separate tasks allows for sharing regulatory commonalities while respecting experimental variance. (B) The number of shared edges between the two datasets, for both model organisms (i) and (ii), shows a high number of overlapping edges. Edges are ranked by their corresponding variance explained for each of the three different model selection approaches: AMuSR, BBSR, and StARS-LASSO. (C) Across the three different model selection approaches, AMuSR learns the highest number of overlapping edges between the respective datasets for model organisms (i) and (ii).



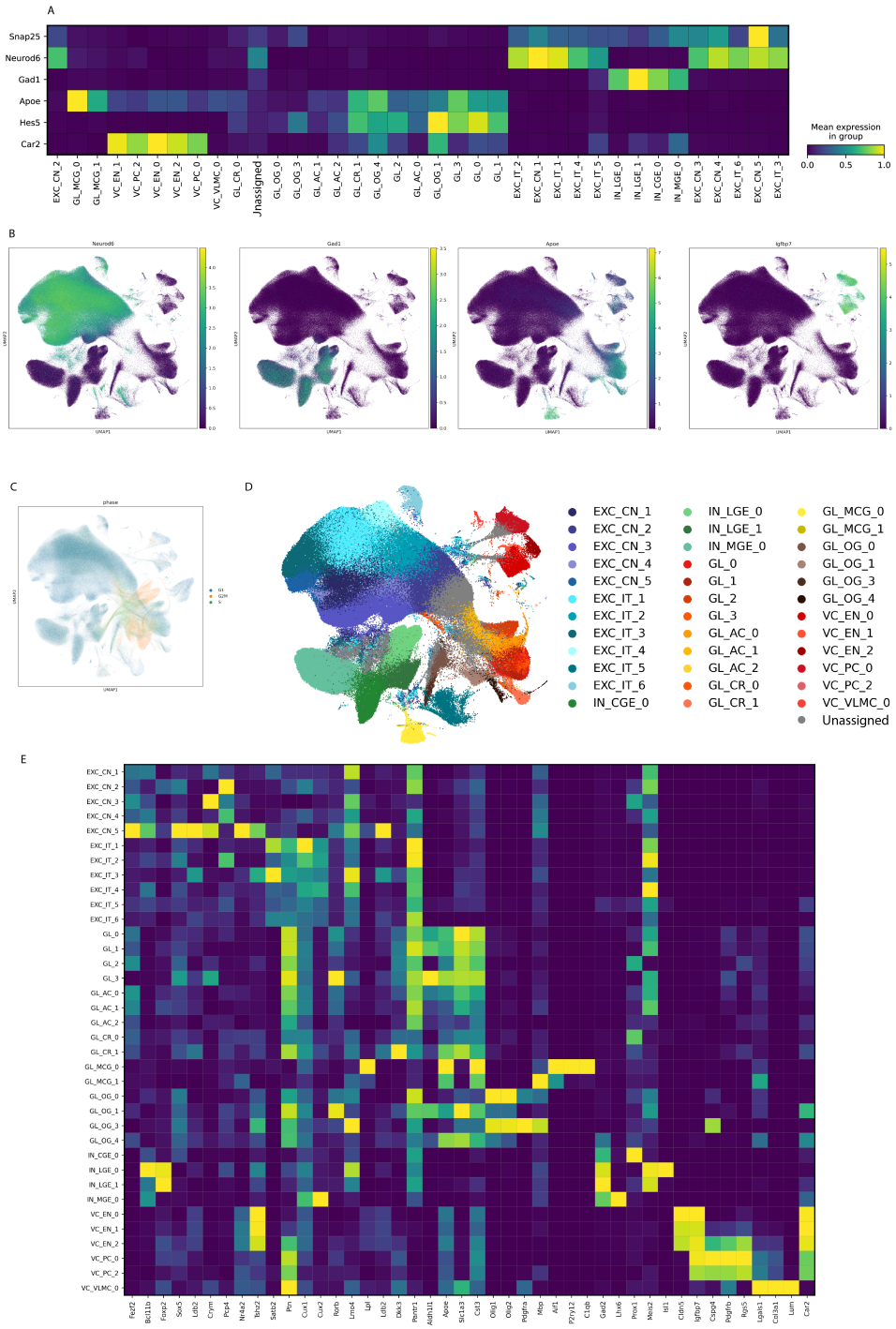
Supplemental Figure 2: Network construction using TF motifs in Saccharomyces cerevisiae. (A) Motifs annotated for GAL4 in the CIS-BP motif database. (B) Histogram of scores linking GAL4 to target genes. Genes in black have been omitted from the final connectivity matrix, and genes in red have been included. (C) Network connecting GAL4 and target genes. Green edges are present in the YEASTRACT database. (D) Histogram of out degree for each TF in the complete network. (E-H) Network analysis as A-D for the JASPAR motif database. (I-L) Network analysis as A-D for the TRANSFAC PRO motif database.



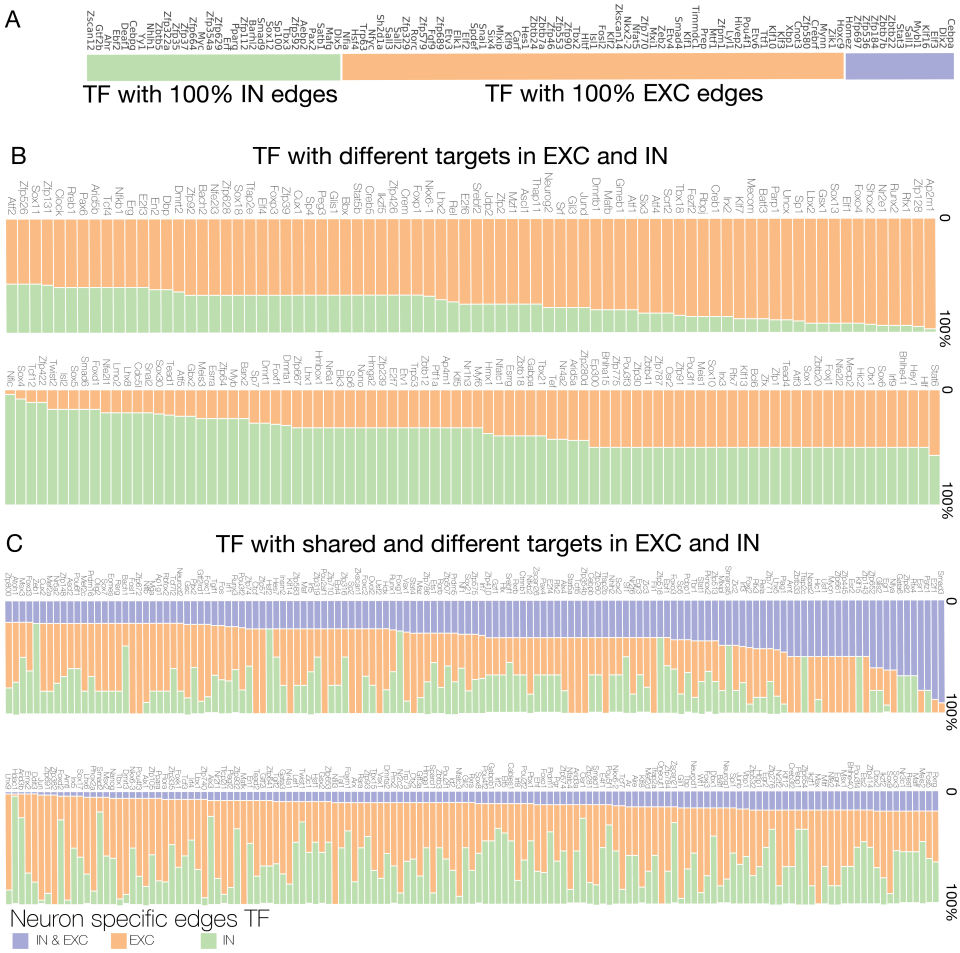
Supplemental Figure 3: Single-cell yeast network inference performance measured by Matthews Correlation Coefficient (MCC) for (A) BBSR, (B) StARS-LASSO, and (C) AMuSR. (D-F) Performance measured by F1 score as A-C



Supplemental Figure 4: (A) The Inferelator 3.0 computational performance as measured by runtime in seconds for BBSR and StARS-LASSO using the Dask engine (140 cpu cores) or using the python-based multprocessing (MP) engine (28 cpu cores). Expression data is sampled from 144,000 mouse cells and 9,782 genes are modeled for network inference. Runtime is shown for 10 replicate runs for each quantity of cells. (B) Number of cells removed during preprocessing for Quality Control (QC), as Mitotic, and as Ambiguous by neuronal marker. Post-mitotic, non-ambiguous cells are retained and clustered. (C) Number of single-cell counts per cell in each of 36 cell type-specific groups, and in the groups removed during preprocessing. (D) Number of genes per cell in each of 36 cell type-specific groups, and in the groups removed during preprocessing



Supplemental Figure 5: (A) Cell class marker expression for each annotated subcluster in mouse single-cell brain data. (B) UMAP of 766,402 mouse brain cells colored by cell class marker expression. (C) UMAP of 1.3M mouse brain cells colored by the assigned cell cycle phase. (D) UMAP of 766,402 mouse brain cells colored by 36 assigned subcluster. (E) Cell type marker expression by assigned subcluster.



Supplemental Figure 6: (A) List of TFs that have identical target genes in GRNs for both Excitatory neurons (EXC) and Interneurons (IN), that have only target genes in Excitatory neurons, and that have only target genes in Interneurons. (B) List of TFs that have no shared target genes in GRNs for Excitatory neurons and in GRNs for interneurons. (C) TFs that have some shared target genes in GRNs for Excitatory neurons and interneurons, but also have some target genes specific to Excitatory neurons or interneurons.