Structure primed embedding on the transcription factor manifold enables transparent model architectures for gene regulatory network and latent activity inference

Andreas Tjärnberg[1,2,3,8,*] **Maggie Beheler-Amass**[2,3,†] **Christopher A Jackson**[2,3,†]

Lionel A Christiaen[1,3,9,10] **David Gresham**[2,3] **Richard Bonneau**[2,3,4,5,6,7,*]

[1]Center for Developmental Genetics, New York University, New York 10003 NY, USA
[2]Center For Genomics and Systems Biology, NYU, New York, NY 10008, USA
[3]Department of Biology, NYU, New York, NY 10008, USA
[4]Flatiron Institute, Center for Computational Biology, Simons Foundation, New York, NY 10010, USA
[5]Courant Institute of Mathematical Sciences, Computer Science Department, New York University, New York, NY 10003, USA
[6]Center For Data Science, NYU, New York, NY 10008, USA
[7]Prescient Design, a Genentech accelerator, New York, NY, 10010, USA
[8]Mortimer B. Zuckerman Mind Brain Behavior Institute, Columbia University, New York, NY, 10010, USA
[9]Sars International Centre for Marine Molecular Biology, University of Bergen, Bergen, Norway
[10] Department of Heart Disease, Haukeland University Hospital, Bergen, Norway

[†]These authors contributed equally.
[*]Correspondence:
andreas.tjarnberg@fripost.org and bonneau.richard@gene.com

Abstract

The modeling of gene regulatory networks (GRNs) is limited due to a lack of direct measurements of regulatory features in genome-wide screens. Most GRN inference methods are therefore forced to model relationships between regulatory genes and their targets with expression as a proxy for the upstream independent features, complicating validation and predictions produced by modeling frameworks. Separating covariance and regulatory influence requires aggregation of independent and complementary sets of evidence, such as transcription factor (TF) binding and target gene expression. However, the complete regulatory state of the system, *e.g.* TF activity (TFA) is unknown due to a lack of experimental feasibility, making regulatory relations difficult to infer. Some methods attempt to account for this by modeling TFA as a latent feature, but these models often use linear frameworks that are unable to account for non-linearities such as saturation, TF-TF interactions, and other higher order features. Deep learning frameworks may offer a solution, as they are capable of modeling complex interactions and capturing higher-order latent features. However, these methods often discard central concepts in biological systems modeling, such as sparsity and latent feature interpretability, in favor of increased model complexity. We propose a novel deep learning autoencoder-based framework, *StrUcture Primed Inference of Regulation using latent Factor ACTivity* (SupirFactor), that scales to single cell genomic data and maintains interpretability to perform GRN inference and estimate TFA as a latent feature. We demonstrate that SupirFactor outperforms current leading GRN inference methods, predicts biologically relevant TFA and elucidates functional regulatory pathways through aggregation of TFs.

1 Introduction

Transcription factor (TF) regulation of mRNA transcription is a main mechanism through which cells control gene expression and respond to context-specific signals [1, 2]. The relationship between TFs and the genes they control forms an interconnected gene regulatory network (GRN), and interpreting this network is necessary to understand cell and organism heterogeneity, development & differentiation, tissue organization, and diseases [3–6]. GRNs are typically represented as causal graphs, which have regulatory TFs linked to target genes. Regulatory interactions in the GRN are difficult to collectively determine experimentally, so it is necessary to computationally infer the structure of the GRN. This inference is complicated by regulatory relationships between TFs and genes that are context-dependent [7], as TFs may control gene expression differently in different cell types or under different conditions [8].

TFs themselves can be regulated transcriptionally (by changing their mRNA levels), translationally (by changing the amount of protein produced from mRNA), or post-translationally (by modifying the protein to alter localization or DNA binding ability). Transcription factor activity (TFA), the relative ability of a TF to alter the expression of target genes, is the aggregate combination of all TF regulation, and is difficult to measure experimentally. Inferring TFA as a latent model parameter is a core component of several GRN inference methods [9–13]. Generally, TFA is inferred from existing evidence of TF to target gene regulation, combined, using linear models, with the measured expression of the target genes. Although powerful, this framework lacks the flexibility to account for heterogeneity and contextual relations observed in biological systems, and the activity estimates have limited interpretability. Workarounds to contextualize regulatory relationships have been proposed [14, 15], however the inflexibility of the models and the lack of interpretability of latent factors remains an issue.

Using more complex models to better match known transcriptional regulatory biology places numerous demands on optimization and inference machinaries and limits scale; using scalable learning and optimization methods from the deep learning field to meet these needs is an appealing way to infer GRNs. They have been used to model expression and covariance networks [16], to build a sparse representation of gene clusters [17], to group genes into co-regulated modules [18, 19], to do supervised clustering of gene sets [20, 21] and for dimensionality reduction and denoising[22]. However, interpretable deep learning models for gene regulation, which provide biological insight into causal relationships in addition to prediction, have been difficult to construct [23]. Techniques to interpret deep learning latent features often focus on removing latent features to quantify their influence and importance [24]. Feature quantification can use change in mean squared error on removal of input features (COM) [25], backpropagation of the output layer into latent features (Grad-CAM) [26], or forward propagation of the latent layers into the output layer [19]. These techniques cannot be used for model pruning or selection, and are not bounded or informative for model structure evaluation.

Knowledge priming embeds existing evidence into the model structure by constraining connectivity between features. This informs the model of constraints which are not directly measured in the training data. These constraints both improve model selection and strengthen interpretability, as priming with biological interactions allows biological interpretation of the resulting network graph [27]. However, this approach lacks the ability to infer novel regulatory structure and is limited to what has been previously documented, as well as to data sets where all required evidence types are extant. Most target genes are regulated by a limited number of TFs that have no direct effect on other genes, encapsulated in the concept of sparsity in GRNs[28]. Model selection, choosing a limited set of regulators per gene, is therefore a key problem for GRN inference, and many techniques have been applied [29–37]. Deep learning models generally remain overparameterized, making biological interpretation difficult, and techniques must be applied after training to eliminate model parameters and enforce sparsity.

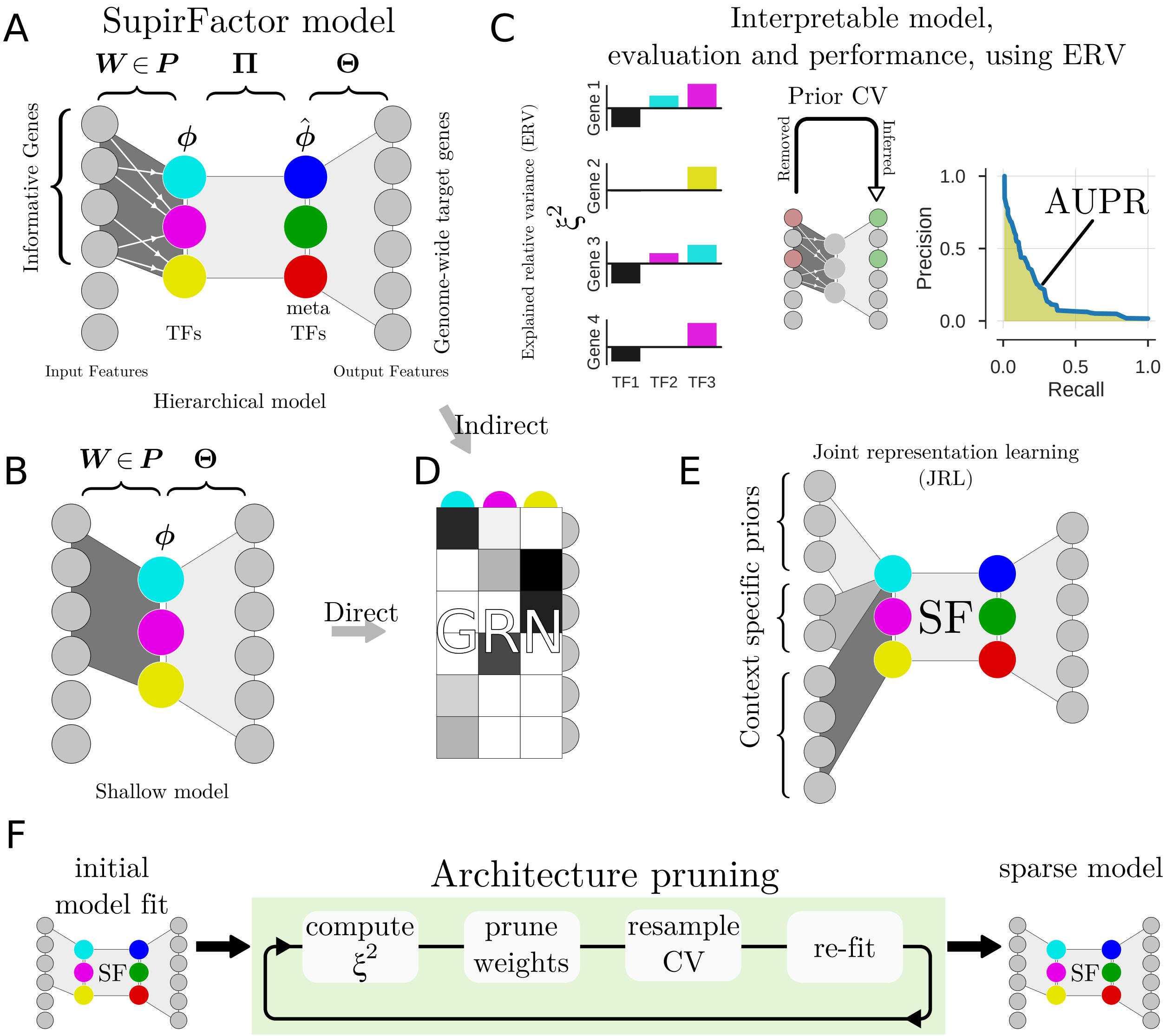


Figure 1: Outline of the SupirFactor framework. The SupirFactor model is constructed like an autoencoder where we embed gene expression data on the transcription factor manifold, exploring two architectures, the "hierarchical" **A**, and the shallow **B** architecture. The output of the first layer defines the latent features marked as TFs (Transcription Factors) and the activation $\boldsymbol{\phi}$ is the transcription factor activity (TFA). The prior $\boldsymbol{P}$ connect the evidence of TF to a set of informative downstream genes, with learnable weights $\boldsymbol{W}$. For **A**, $\boldsymbol{\Pi}$ connects the TFs to the latent features, here called the meta TFs (mTFs). $\boldsymbol{\Theta}$ weights the mTF activity (mTFA) to predict genome wide gene expression profiles. In **B** the TFs directly weights TF to gene influence in $\boldsymbol{\Theta}$. **C**: To make the model completely interpretable and transparent we use explained relative variance (ERV) ξ^2. ERV estimate importance of all latent factors influence on model output features. This is then used to evaluate the model and its performance. The GRN is cross validated, where genes to TF connections are held out in the input $\boldsymbol{W}$ and predicted in the GRN which for the shallow model is $\boldsymbol{\Theta}$ and for the hierarchical model is $\check{\boldsymbol{\Theta}}$ the indirect effect from the TFs to output features. The measured recovery of these links gives insight on stability and biological relevance of the GRN where parameters are ranked by their predictability measured by ξ^2. **D**: Gene regulatory network extracted as indirect TF-gene interaction in hierarchical SupirFactor and direct TF-gene interactions in shallow SupirFactor. **E**: Multi-task learning is implemented in SupirFactor through a joint representation learning (JRL) architecture where biological distinct contexts is independently weighted into a joint GRN representation. **F**: Architecture pruning and sparsity procedure in SupirFactor is used to stabilise and eliminate over-parameterization by eliminating non predicting model parameters facilitated be ERV.

2 Results

We present a model inspired by white-box machine learning approaches [38], that we call *StrUcture Primed Inference of Regulation using latent Factor ACTivity* (SupirFactor). This model incorporates knowledge priming by using prior, known regulatory evidence to constrain connectivity between an input gene expression layer and the first latent layer, which is explicitly defined to be TF-specific. This model ties the latent TF layer causally to *informative genes*, and allows this layer to be directly interpreted as transcription factor activity (TFA). This latent layer is then linked to gene expression in an output layer, which is interpreted as an explicitly inferred GRN.

We also adapt new metrics for model interpretation in this context, we define explained relative variance (ERV), a novel concept to interpret the structure of the inferred network graph for any architecture. Briefly, ERV is defined as the change in residual variance when a latent feature is removed from the model, and is used to rank and interpret graph weight importance within the model. Using ERV allows TF to gene interactions to be interpreted through additional latent layers placed between the TF latent layer and the output layer.

Benchmarking across multiple datasets we find that SupirFactor outperforms previous methods using similar frameworks for recovering GRNs. We find that our model uncovers biologically-relevant TFA and predicts biological function of latent aggregates of TFs in deeper layers, suggesting our model is useful for predictive analyses beyond inferring GRNs. In particular we expect to predict activity of specific TFs and to aggregate TFs into regulatory pathways, that we demonstrate on an experimental *S. cerevisiae* data set and a mammalian large single cell PBMC dataset. GRN interpretability and context specific network analysis is facilitated by using ERV and, we demonstrate its utility by applying trained models to context specific and unseen data.

2.1 The SupirFactor Regulation Model

The SupirFactor model learns a set of weights connecting genes to TFs in the prior, where these genes functionally serve as reporters for the activity of connected TFs. As opposed to inferring TFA explicitly, as in network component analysis (NCA) (Section 4.4.4) where regulatory evidence is static in the prior, we set $\boldsymbol{\phi} = g(\boldsymbol{W}, \boldsymbol{x})$, where $\boldsymbol{W} \in \boldsymbol{P}$ is the weighted influence of genes to TFs derived from prior evidence $\boldsymbol{P}$, and $\boldsymbol{x}$ is the expression of genes informing on $\boldsymbol{\phi}$ which may be a subset of *informative genes* connected through $\boldsymbol{W}$. This extends our model to $\boldsymbol{x} = f(g(\boldsymbol{W}, \boldsymbol{x}), \boldsymbol{\Theta})$, where $\boldsymbol{\Theta}$ is the GRN.

We actualise this model using a deep learning framework choosing f, and g and learn $\boldsymbol{W}$, and $\boldsymbol{\Theta}$ with $\boldsymbol{\phi}$ as a latent feature prduced by the weighted output from $\boldsymbol{x}$ mapped through a learned $\boldsymbol{W}$. Depending on the form of f, and including non-linearities, we can learn additional higher order interactions and regulatory pathways (Section 4.1). The complex version of the model that can capture other interactions in f we call "hierarchical" SupirFactor (Figure 1A). A simple version of this framework uses a single bottleneck layer for our function f we refer to as "shallow" SupirFactor (Figure 1B).

Constructing a prior matrix $\boldsymbol{P}$ is a challenging but essential task for including informative evidence of regulation. This step is also a way to integrate data types that can shed light on TF-target relationships. This matrix can represent previously known interactions, and it can also encode higher probability interactions derived from chromosome accessibility or TF-chromatin interactions (experimentally measured by ATAC-seq and ChIP-seq) [39, 40]. A more dense $\boldsymbol{P}$ is likely to include more false positives and will therefore result in an noisier propagation of TF variance. A sparser $\boldsymbol{P}$ is likely to have many false negatives, limiting the variance that the model is able to explain, resulting in an model that may be less predictive.

A concern is that prior connectivity $\boldsymbol{P}$ rarely includes reliable sign or weight estimates. Inferring signs for $\boldsymbol{P}$ from the direction of change after perturbations is technically difficult, as it requires perturbing all TFs included in the model. Relying on expression correlation to infer signs will conflate both indirect and co-varying regulation. We expect that refitting weights $\boldsymbol{W} \in \boldsymbol{P}$ dependant on $\boldsymbol{\Theta}$ will mitigate these problems.

2.2 Selection of SupirFactor hyperparameters

We evaluate the SupirFactor framework (Figure 1) for GRN inference from bulk and single-cell RNA expression data. First, to test our model setup, explore interpretability, and compare performance to other models, we benchmark using multiple data sets where a partial ground truth network is available (called the gold standard in this work) on the "shallow" SupirFactor. We have previously assembled a GRN inference data package that consists of two prokaryotic *Bacillus subtilis* bulk RNA expression data sets (B1 and B2), two

Saccharomyces cerevisiae bulk RNA expression data sets (S1 and S2), and one *S. cerevisiae* single-cell RNA expression data set (scY)[41]. This data package also includes a *B. subtilis* gold standard network [10] and a *S. cerevisiae* gold standard network [42], both derived from literature databases. Single-cell RNA expression data is preprocessed (Section 4.7.1) and both bulk and single-cell data is feature normalized (Section 4.7.2) prior to model fitting.

To be able to extract a GRN from our model that gives us latent feature to output connections, we need to be able to interpret model weights, features and their relative importance. This is difficult in multi-layer architectures, and weights may not be scaled with biological importance. Instead, we devise a metric to quantify the direct effect a latent feature has on its targets in the output, Explained Relative Variance (ERV) ξ^2 (Figure 1C), with appealing properties for GRN inference. This metric is based on feature removal [24], and is computed as the coefficient of partial determination (CPD) [43]. We silence each latent model feature and compute the consequent effect on the output variance MSE, scoring the silenced model feature against the full model prediction without retraining (Section 4.4.1).

Deep learning models have a number of hyperparameters that needs to be tuned for optimal model performance. Two critical hyperparameters for GRN inference are the *weight-decay*, typically called the L2 penalty, and dropout, stochastic perturbation of the data during training to attenuate noise and improve model generalisation (Section 4.3). We test dependence on these hyperparameters by searching for L2 and dropout (on input and latent features) with the simplified shallow SupirFactor model. Each hyperparameter value is tested by splitting expression data 50-50 into training and validation set, using 80% of the gold standard network as prior network information for the model and holding out 20% for scoring. Negative controls consist of either shuffling the data or the prior network. Area under precision recall curve (AUPR) is used to score the network structure against the gold standard network, and R^2 is used to quantify prediction accuracy. Each configuration is rerun repeatedly and average performance is reported (Supplemental Figure S2-S9).

We observe that in some cases (for S2 and B1 datasets), when increasing L2 beyond a specific value, R^2 decreases while AUPR increases (Supplemental Figure S3 & S4). We interpret this as overfitting to the prior network structure and increasing recovery of the gold standard in cases where the these two structures align, such as in scale-free networks with dense highly connected TFs. The model then emphasises these TFs at the cost of prediction accuracy and inclusion of less connected TFs. This is undesirable, so we select hyperparameters that maximises R^2 while maintaining a high AUPR. Negative controls perform as expected; shuffling the data eliminates the biological interpretability and predictive power of the resulting GRN, and shuffling the prior network eliminates only biological interpretability while still achieving good predictive power.

To determine an optimal L2 we look for where R^2 is maximized. We find that prediction accuracy is maximised in the span $R^2 \in 10^{-6} - 10^4$ for all data sets, where we select an L2 of 10^{-4} for B1 and B2 and 10^{-6} for S1, S2 and scY for further comparisons (Supplemental Figure S2C-S9C). For dropout, in general, we find that setting larger dropout on input and smaller dropout on latent features increases AUPR while maintaining a higher R^2.

2.3 SupirFactor benchmarking demonstrates improved biological regulatory network recovery

We summarize the performance of the shallow SupirFactor model using a linear activation function with several modeling choices (Figure 2A). Using ERV as an estimator for biological relevance outperforms interpretation based on model weights alone, as determined by AUPR. Selecting the optimal dropout hyperparameters based on maximum R^2 for the selected L2 (max R2) improves model R^2 at a cost of decreasing network prediction performance (AUPR), when compared to setting a fixed dropout (fixed; input=0.5 and latent=0).

For single-cell RNA expression data (Figure 2B) we extend the comparisons, comparing linear and rectified linear (ReLU) activation functions (Section 4.8), and comparing standard normalization to a robust sparsity-retaining normalization (Section 4.7.2). RobustMinScaler outperforms StandardScaler normalisation, implying that preserving sparse data for model training is advantageous (Figure 2B). To summarize these benchmarking results, ERV should be used to evaluate model parameters, and dropout hyperparameters can be fixed without loss of prediction accuracy and GRN recovery.

Finally, we compare shallow SupirFactor performance to that of the Inferelator (Figure 2C), a method which takes comparable RNA expression and prior network inputs and learns a GRN. We note that other methods

Context:		A		B		A	B		
A	B	$\xi^2(A) > \xi^2(B)$	$\xi^2(B) > \xi^2(A)$	$\xi^2(A) > \xi^2(B)$	$\xi^2(B) > \xi^2(A)$	Total	Total	OR	p-val
G1	S	589	152	83	196	1881	465	9.2	6.0e-49
S	G2	104	181	62	217	465	305	2.0	2.2e-04
G2	M	108	166	109	652	305	1452	3.9	6.4e-17
M	M/G1	549	232	96	222	1452	717	5.5	2.0e-34
M/G1	G1	583	184	111	222	717	1881	6.3	8.7e-41
G1	G2	623	132	176	108	1881	305	2.9	1.3e-11
S	M	591	197	43	207	465	1452	14.4	8.9e-61
G2	M/G1	104	171	74	262	305	717	2.2	2.4e-05
M	G1	582	190	192	584	1452	1881	9.3	5.5e-92
M/G1	S	238	97	62	212	717	465	8.4	8.3e-34

Table 1: Pairwise comparisons between context-specific cell cycle phase networks from hierarchical SupirFactor. Network edges are classified into cell cycle phase context based on the existing annotation of the target gene. ERV of the network edges (where $\xi^2 > 0.01$ in one or more contexts) is then compared as a contingency table. Network edge counts in gray columns are edges where ERV is higher in the correct context-specific network, based on existing annotations. The total column counts network edges (where $\xi^2 > 0$ for that context) for the context-specific network, and does not total the columns of the contingency table. Odds Ratio (OR) and p-values are calculated by one-sided fisher exact test, using the contingency table.

which take these inputs are not amenable to scoring on holdouts that we are employing for our benchmark [41], making comparison difficult. Using the configurations from our benchmarking, SupirFactor improves significantly over previous results on all of the data sets tested. Comparison of the Inferelator to several alternate methods on the same data sets used here has been previously shown [41].

2.4 Hierarchical SupirFactor with non-linear activation facilitates interpretation of latent feature activation as biological activity in cell contexts

To account for more complex, multi-TF regulatory interactions we extend our mode in multiple ways, introducing non-linearities and adding additional latent layers that represent interactions. Non-linear functional forms are necessary in the proposed hierarchical SupirFactor architecture to model interactions more complex than linear relationships. This is necessary as GRNs are context- and cell-type-dependent; and thus a TF to gene regulatory interaction may *e.g.* exist when the organism is in one state, but be inaccessible in another.

SupirFactor can distinguish contextual networks by embedding context specific assigned data and computing ERV only within that data set. We explore learning context-dependent GRNs here; we evaluate this ability on the single-cell RNA expression data set, which has samples annotated by growth conditions (Figure S1A). We compare hierarchical SupirFactor (Figure 1A), shallow SupirFactor (Figure 1B), and a comparable multi-task learning approach (AMuSR) in the Inferelator that also learns context-dependent networks. Both the shallow SupirFactor and hierarchical SupirFactor outperform the Inferelator (Figure 2D). Shallow SupirFactor outperforms the hierarchical SupirFactor in some contexts, although the shallow model uses a linear activation function, and the hierarchical model uses a ReLU activation function. As this activation function constrains the latent features to be strictly positive, latent features are interpretable in the hierarchical model.

We use hierarchical SupirFactor to construct context-specific GRNs for cell cycle phases, by inferring cell cycle phase from transcriptional markers (Figure S1B). Regulatory edges that are actively used in a context-specific GRN should explain more relative variance, compared to edges which are inactive. For each cell cycle we used the gene annotation of cell cycle phase genes and compare ERV between phases for each relevant gene (Table 1). We compare both neighbouring phases and phases skipping the immediate following phase. Computing a one sided fisher exact test classifying genes in the GRN where they have the highest ERV, only using ERV that has $\xi^2 > 0.01$ in at least 1 of the conditions, we find that for all comparisons we have a strong enrichment of the phases relevant genes in terms of ERV in the corresponding phases.

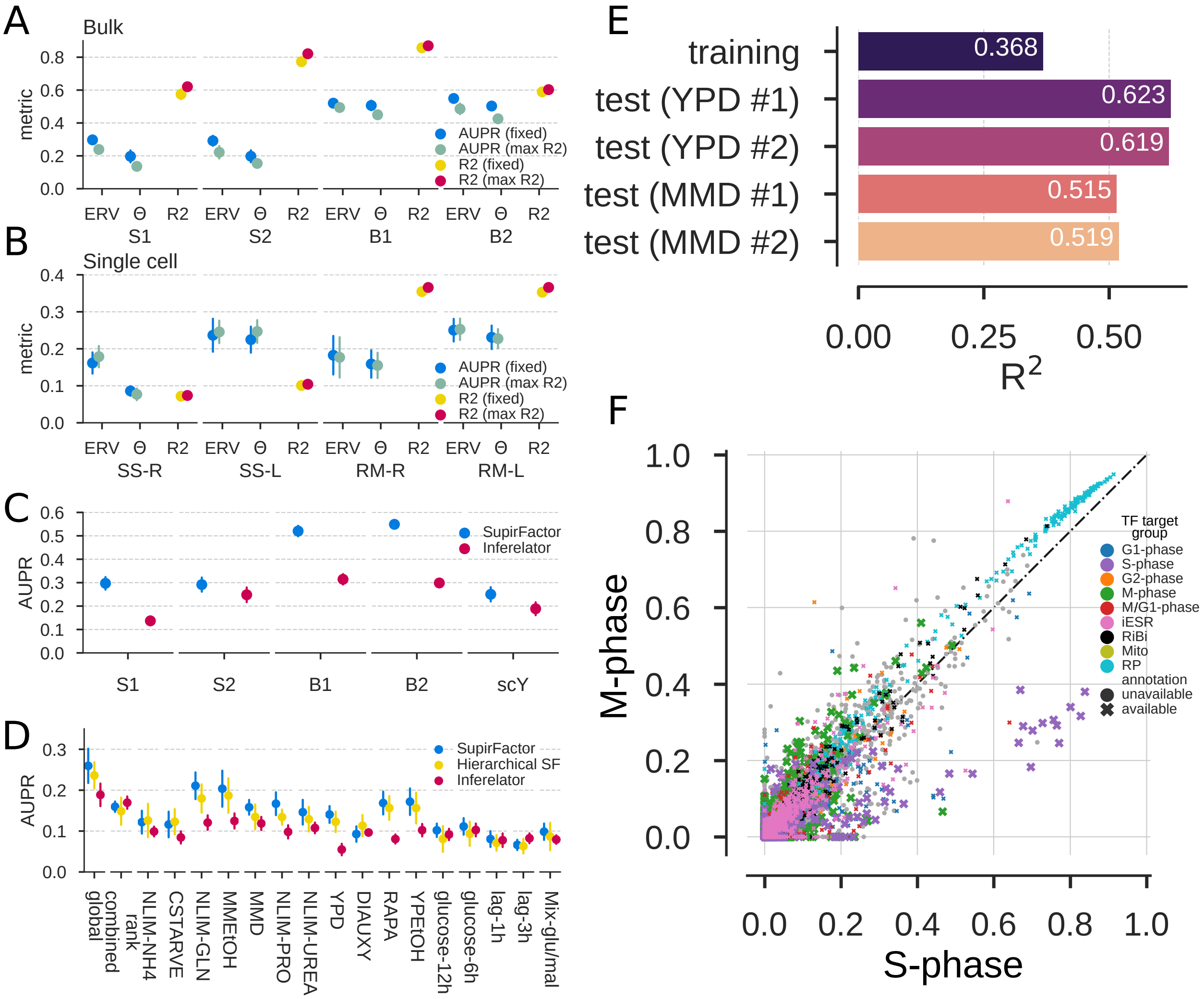


Figure 2: SupirFactor benchmark and hyperparameter evaluation. Performance is evaluated on gold-standard networks consisting only of edges held out of the prior network $\boldsymbol{P}$, measuring recovery using AUPR. R^2 is computed on a validation set of 50% of the data samples held out of the training data. Data sets are labeled for species, with *B. subtilis* (B1 and B2), *S. cerevisiae* bulk RNA expression (S1 and S2), and *S. cerevisiae* single-cell RNA expression (scY). **A**: Comparing network interpretation using model weights (Θ) to interpretation using Explained Relative Variance (ERV), measured using AUPR against edges held out of the prior network $\boldsymbol{P}$. R^2 is calculated from the full network model. Model hyperparameters are set based on Supplemental Figure S2-S9 and Section 4.3. **B**: Comparing normalization and activation functions for single-cell RNA expression data, as in (**A**). SS- is normalizing to mean of zero and unit variance, RM- is normalizing to maintain a minimum value of 0 (retaining sparsity), -L is linear activation, and -R is ReLU activation. **C**: Benchmarking SupirFactor with optimal parameters selected from (**B**) against a comparable GRN inference method, the Inferelator. **D**: Comparing multi-context network performance between shallow SupirFactor, Hierarchical SupirFactor, and the multi-task Inferelator. GRNs are learned from single-cell (scY) data, with context/task groupings determined by growth condition. Global GRNs are learned from the data without separate groupings (using StARS-LASSO for the Inferelator [41]). Context networks are computed post-training in SupirFactor and split here on growth condition. **E**: Evaluating model prediction R^2 on four novel test data sets, using a GRN trained by Hierarchical SupirFactor. **F**: Comparing contextual network for GRNs defining cell cycle M-phase and S-phase (Table 1). Each point is an interaction from the two contextual networks, colored by the target gene functional annotation. X and Y axis are ξ^2 of the S-phase and M-phase networks. GRN interactions targeting S-phase genes (purple) have higher ERV in the S-phase contextual network, and interactions targeting M-phase genes (green) have higher ERV in the M-phase contextual network.

Hierarchical SupirFactor introduces a larger set of learnable weights and a potential over-parameterisation of the model, and, thus, the expanded model presents new model selection challenges in the context of sparse GRN inference. We test if an iterative method for removing model weights could be used without loss of model performance. ERV is used to rank model weights, and an $\xi^2 \leq 0$ indicates a non predictive model weight. Identifying these non-predictive links, pruning them, and then refitting the model attenuates over-parameterization (Figure 1E). After 1 iteration of pruning, nonzero weights reduce to $\sim 57\% \in \mathbf{\Theta}$, $\sim 52\% \in \hat{\mathbf{\Theta}}$ and $\sim 80\% \in \mathbf{\Pi}$ of the pre-pruning model trained on scY. It is critical to determine if models learned by hierarchical SupirFactor generalize. We evaluate the predictive ability of a hierarchical SupirFactor model trained on data set scY using prior knowledge $\boldsymbol{P}$ without holdouts (Figure 2B, RM-R). To do this, we generate four new experimental test single cell RNA expression data sets by collecting and sequencing cells grown in environmental conditions seen in the training data set (YPD and MMD). This model explains the variance of the training data well ($R^2 = 0.37$), and also explains the variance of the YPD ($R^2 = 0.62$) and MMD test data sets ($R^2 = 0.52$) well (Figure 2E). We conclude that SupirFactor generalises and predicts expression patterns of new data even when model weights have been removed.

The SupirFactor model explicitly fits an intermediate layer which can be directly interpreted as latent Transcription Factor Activity (TFA) for each TF, when this layer uses a ReLU activation function [44]. Using hierarchical SupirFactor, we calculate latent TFA for all TFs. When examining the role of cell cycle TFs, the advantages of TFA are apparent. The TFA for cell cycle TFs is maximal in phases the TFs are expected to regulate (Figure 3A), based on known TF roles from literature[45]. Almost all cells have non-zero TFA for cell cycle TFs in at least one phase of the cell cycle, but TF expression is highly sparse, complicating causal linkage to targets based on TF expression (Figure 3B). We note that for these cell cycle TFs, the expression of the TF often peaks in the phase before the TFA of the TF.

We also wanted to test the role of functionally related TFs. Transcription factors often interact with each other to define regulatory states, as part of multisubunit complexes, or by competing for the same DNA binding regions. These interactions should be reflected in how TF activity correlates, even if these interactions are not explicitly embedded in the model. We select several typical examples of TF pairs with known interactions and compare inferred TFA between hierarchical SupirFactor and the Inferelator (Figure 3C). MSN2 and MSN4 are partially redundant stress response TFs that bind to the same DNA motif, and we expect their activity to be partially correlated. Hierarchical SupirFactor TFA is weakly correlated ($r = 0.18$), unlike expression of MSN2 and MSN4, which are uncorrelated ($r = 0.01$). RTG1 and RTG3 are obligated to form a physical dimer for functionality, and we expect their activity to be strongly correlated. Hierarchical SupirFactor TFA is strongly correlated ($r = 0.61$), unlike expression of RTG1 and RTG3, which are uncorrelated ($r = 0.01$). Finally, HAP2 and HAP5 are part of the multisubunit heme-activated TF complex, and we expect their activity to be strongly correlated. In this case, hierarchical SupirFactor is less successful at correlating TFA ($r = 0.14$) than the Inferelator ($r = 0.33$), expression is again uncorrelated ($r = 0.03$). Overlaying TFA onto a reduced-dimensionality plot allows for the comparison between TF activities and the experimental conditions which cause them to be correlated or uncorrelated (Figure 3D).

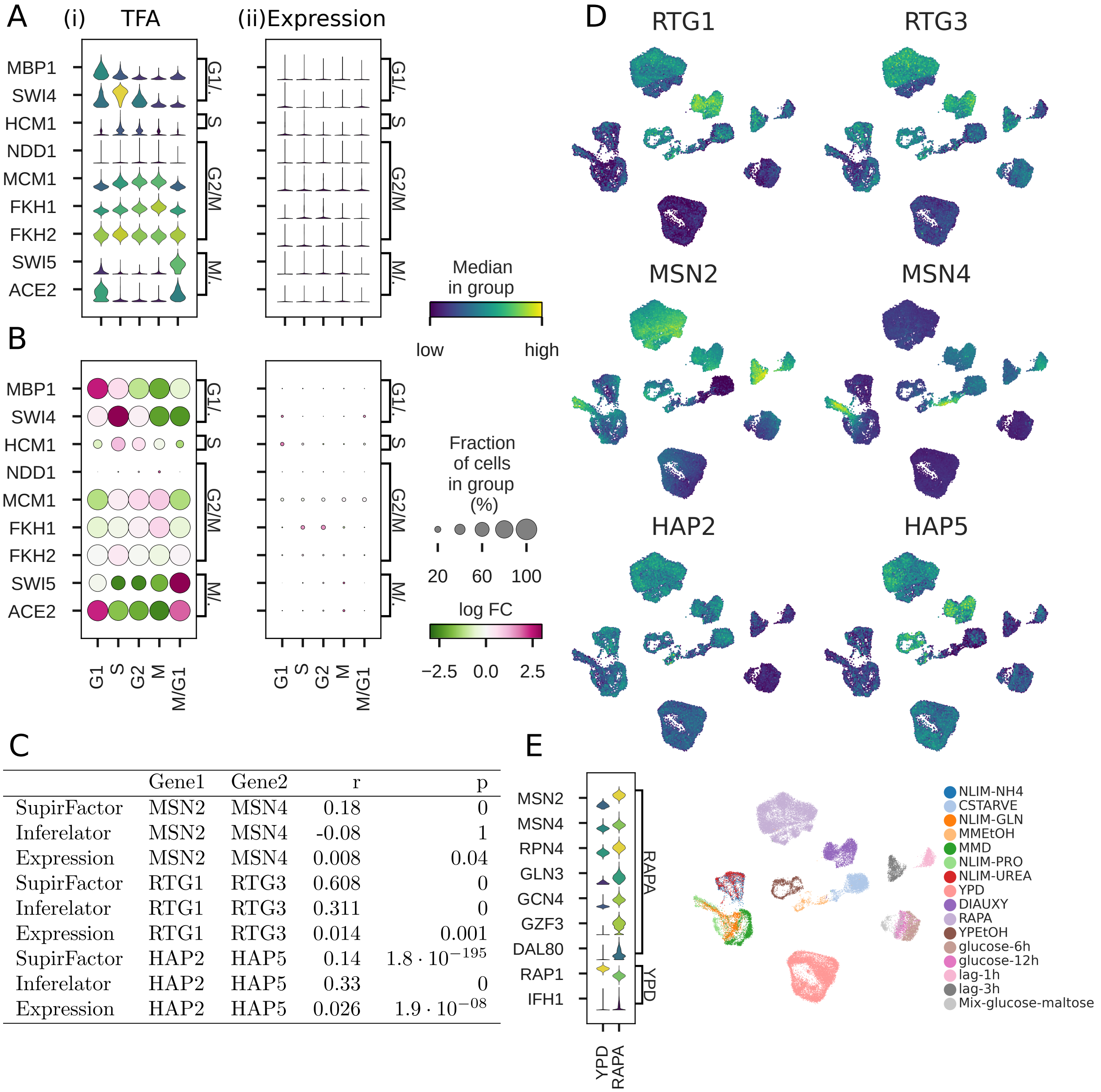
Finally, we compare the TFA between perturbed, rapamycin-treated cells and untreated control cells (Figure 3E). Rapamycin is expected to inhibit TOR pathway signalling, altering stress response and nutrient response TF activities [45]. By comparing the TFA between perturbed and control cells, hierarchical SupirFactor is able to reconstruct which TFs are activated and deactivated by this perturbation.

2.5 Hierarchical SupirFactor combines TFs into pathways

In hierarchical SupirFactor we introduce an additional latent layer, which we interpret as meta transcription factors (mTFs) that aggregate TFs into multi-regulator pathways. As this mTF layer is directly connected to the output gene expression, we expect that the mTF layer activity (mTFA) can be interpreted as the activity of a regulatory pathway.

To test this hypothesis, we explore the hierarchical SupirFactor model trained on the single-cell yeast data (scY). mTF functions are determined by enrichment for regulation of genes that are annotated with Kyoto Encyclopedia of Genes and Genomes (KEGG) pathways [47]. As this data was collected from cells growing in different carbon and nitrogen sources, we focus on enrichment of a specific subset of metabolic pathways. 42 of 148 mTFs are enriched for target genes (defined as mTF to gene connections with $\xi^2 > 0.1$) to these metabolic KEGG pathways (Figure 4A).

We focus on the 12 mTFs which are enriched for glycolysis target genes, a pathway which is a core part of the central carbon metabolism. Many of the growth conditions in the training single-cell yeast gene expression



	Gene1	Gene2	r	p
SupirFactor	MSN2	MSN4	0.18	0
Inferelator	MSN2	MSN4	-0.08	1
Expression	MSN2	MSN4	0.008	0.04
SupirFactor	RTG1	RTG3	0.608	0
Inferelator	RTG1	RTG3	0.311	0
Expression	RTG1	RTG3	0.014	0.001
SupirFactor	HAP2	HAP5	0.14	$1.8 \cdot 10^{-195}$
Inferelator	HAP2	HAP5	0.33	0
Expression	HAP2	HAP5	0.026	$1.9 \cdot 10^{-08}$

Figure 3: Transcription factor activity in single cell yeast. TFA estimated from hierarchical SupirFactor model. Violin plots are generated by scaling [0, 1] the underlying measurement. **A**: Cell cycle TFs regulate gene expression in specific cell cycle phases, with the phase the TF regulates annotated on the right y-axis [45]. Cell cycle phase of each cell is inferred from transcriptome, and annotated on the x-axis. Panel (i) plots TFA, and panel (ii) plots the RNA expression of the TF. **B**: Same as **A**. Dot size represents the percentage of cells with a non-zero value. Color represents log fold-change (log FC) across the cell cycle phases. Panel (i) plots TFA, and panel (ii) plots the RNA expression of the TF. **C**: Interacting TF pair pearson correlation for SupirFactor TFA, Inferelator TFA, and TF expression **D**: Comparing TFA between rapamycin (RAPA) treated and untreated (YPD) cells TFs known to be activated by treatment [46] or known to be more active in untreated cells are annotated on the right y-axis. **E**: UMAP projection of the scY dataset showing TFA estimate of co-regulator TFs and growth conditions.

data set use glucose as a primary carbon source (Supplemental Figure S1E), and we see that these conditions have very similar mTF activation (Figure 4B). The remaining growth conditions use different carbon sources, requiring different regulation of the central carbon metabolism. Six of the glycolysis mTFs are activated in these non-glucose carbon sources, but have considerably more repressive (negative weighted) links to target genes (Figure 4C), suggesting that they are mainly downregulating glycolytic genes. We can further overlay mTF activity onto a low-dimensionality projection in order to identify mTFs which are linked to carbon source with little heterogeneity (*e.g.* 74 & 79) and which have heterogeneity within growth conditions (*e.g.* 57 & 78) (Figure 4D). We observe that mTFs aggregate biologically functional groups in their targets and can be evaluated quantitatively as activities of these pathways.

2.6 SupirFactor models regulation of mammalian PBMCs with multimodal single cell sequencing data

We evaluate the use of SupirFactor to model complex biological systems by applying the method to model Peripheral Blood Mononuclear Cell (PBMC) gene regulation, using a paired multi-omic single cell ATAC-seq and RNA-seq dataset [48] (Figure 5A). These two data types are integrated by using ATAC-seq chromatin accessibility as a cell-specific mask (Section 16).

PBMCs are a heterogeneous pool of multiple cell types, and each cell type may have subsets of cells in different states. We annotate these cells as dendritic cells (DC), monocytes (Mono), natural killer cells (NK), B-cells (B), CD4$^+$ T-cells (CD4 T) and CD8$^+$ T-cells (CD8 T). To account for this heterogeneity, we build a context aware Joint Representation Learning (JRL) SupirFactor model (Section 4.2). This allows the model to weight regulatory evidence based on context and aggregate into a joint latent feature space to build a joint SupirFactor GRN model. We define cell contexts for JRL by clustering cells using Leiden clustering[49], with a resolution = 0.2, generating 7 clusters (Supplemental Figure S11A-B). The PBMC SupirFactor model was trained using JRL on ATAC masked data (epochs=400) and subset to the most explanatory model weights once (epochs=100), resulting in a model that is scored on held-out cells ($R^2 = 0.35$). The modeled PBMC regulatory network predicts 818 active TFs regulating 13698 genes, connected through 492 mTFs in a latent mTF layer ($\xi^2 \geq 0$) (Supplemental Figure S11C).

Eleven of these mTFs are linked to target genes which are functionally enriched for immune cell specific KEGG pathways (Figure 5B), and we interpret these mTFs as regulatory pathways. As an example, mTF-637 is active in B cells but largely inactive in other PBMCs (Figure 5B-D); the target genes of this mTF are functionally enriched for B-cell receptor signaling (Figure 5B). mTF 313 is activated in dendritic cells (DCs) and regulates genes functionally enriched for phagocytic activity and for Neutrophil extracellular trap formation, which activate DCs and allow them to mature [50]. mTF 227 is also linked to phagocytosis, although is principally active in monocytes. mTF 17 is active in NK cells with functional enrichment for Natural Killer cell mediated cytotoxicity genes. Overall, this demonstrates the utility of SupirFactor mTFs as a tool for identifying cell type specific regulation.

We can further examine TFs that have inferred cell-type specific activity (Figure 5E). SupirFactor distinguishes activation between myeloid (monocytes and DCs) and lymphocytic (B, NK and T cells) lineages (Figure 5E). The framework also recovers known cell-type specific regulators along the myeloid lineage, monocytes and DCs, (KLF1) [51], B cells (PAX5, SMAD5) [52, 53], NK cells (ARNT) [54], CD4 T cells (IRF1, RUNX3) [55, 56], and CD4 T cells (NR4A1) [57]. We show that SupirFactor infers cell-type-specific differential mTF activation and TF activation among distinct cell types that correspond with known biological processes and protein activity for multiple key cell types. The analysis of SupirFactor performance on the PBMC dataset demonstrates that SupirFactor can learn biologically relevant interactions in complex organisms and datasets.

3 Discussion

In this study we describe *StrUcture Primed Inference of Regulation using latent Factor ACTivity* (SupirFactor), a model within the class of knowledge primed deep learning models. SupirFactor explicitly treats transcription factor activity as an interpretable latent state which drives gene transcription. This model uses a single objective function where the influence of the prior regulatory structure is optimized together with the GRN. SupirFactor combines the power of DNN optimization with prior structure constraints for inferring GRNs and explicit estimation of TFA. These TFA estimates are bounded by a ReLU activation function, and are directly quantifiable and interpretable on a per-observation basis.

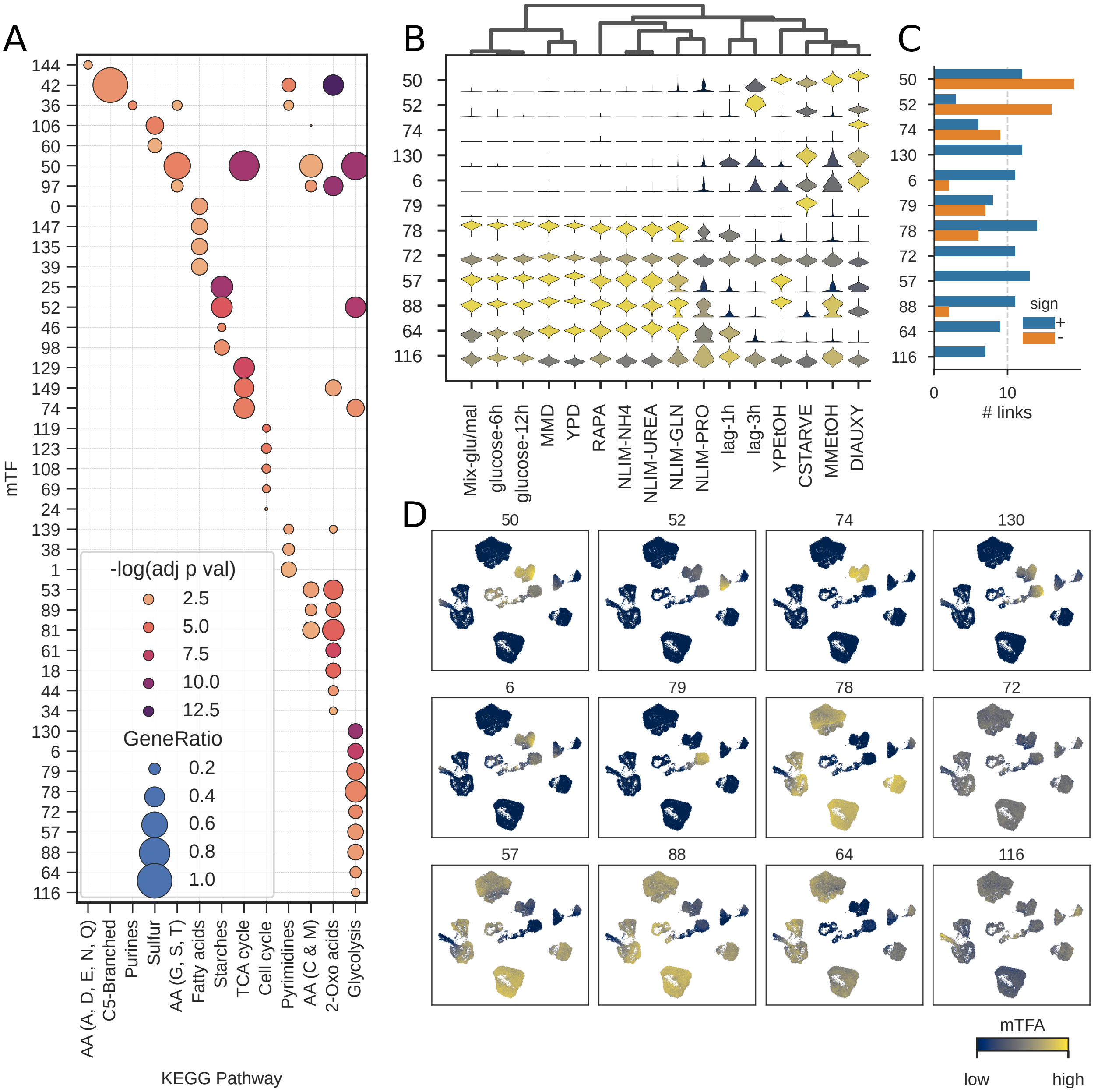


Figure 4: Meta transcription factor (mTF) functional enrichment analysis in single cell yeast. TFA estimated from hierarchical SupirFactor model trained in Figure 3. mTFs are nodes in the SupirFactor model $\check{\Phi}$ latent layer and are numbered from 1-148. **A** Pathway enrichment of mTFs to selected core metabolic KEGG pathway annotations on target genes. **B** mTF activity for cells in each growth condition (mTF activity scaled [0, 1] for comparison). **C** Positive (activating) and negative (repressing) weights from mTFs to target genes within the Glycolysis KEGG-pathway for each mTF. **D** mTF activity for cells overlaid on a low-diensional UMAP projection. Cell metadata plotted in Supplemental Figure S1.

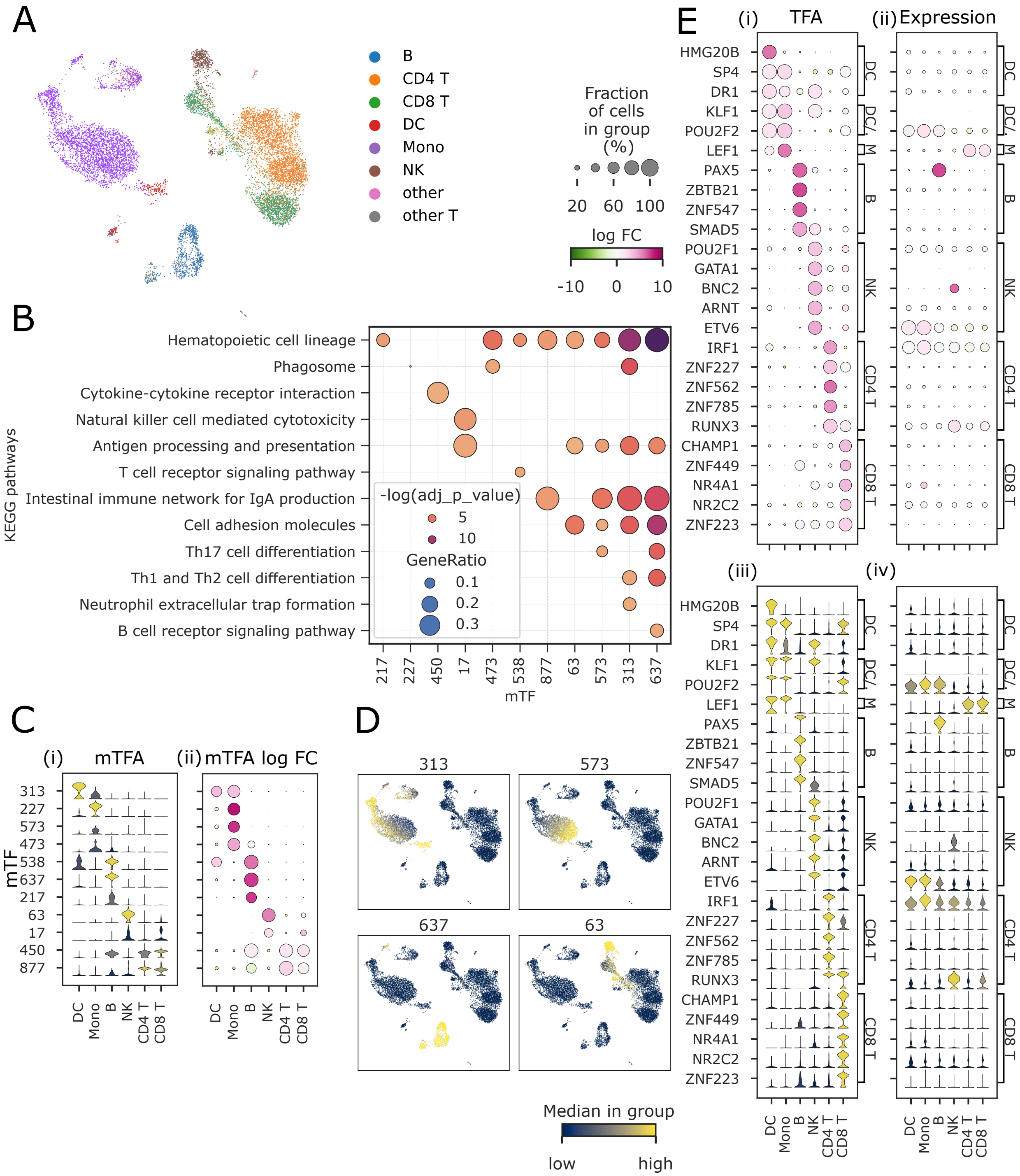


Figure 5: Single cell PBMC TFA and mTF functional enrichment analysis. TFA and mTFA are estimated from a JRL SupirFactor model. mTFs are associated with latent features $\hat{\phi}$ and are numbered 0-881. **A** UMAP projection of the single cell PBMC dataset, labeled with cell type annotations. **B** Selected enriched terms and associated mTFs for cell-type specific KEGG pathway based on mTF target genes, $\xi^2 > 0.01$. **C** Activity of functionally enriched mTFs, over each cell-type. **D** UMAP projection of PBMCs colored by mTF activation. **E** Significant TF activation for specific cell-type populations and corresponding gene expression (scaled [0, 1] for comparison). Dot plots of (i) TFA and (ii) TF mRNA expression. Violin plots of (iii) TFA and (iv) TF mRNA expression.

SupirFactor has been carefully benchmarked using both bulk RNA expression data sets and single cell RNA expression data sets. We rely on model organisms *B. subtilis* and *S. cerevisiae* for benchmarking, as these organisms are well-characterized and have a partial experimentally validated ground truth network available, which we use for scoring recovery of GRN structure. This model organism benchmarking is important, as mouse and human data sets used for GRN inference benchmarking often lack reliable ground truth networks for scoring, and are restricted to using predictive metrics which have limited value. This benchmarking shows that the SupirFactor framework is versatile and has improved GRN inference over a comparable framework that relies on statistical learning, measured by recovery of network edges which are held out of the modeling. The SupirFactor GRN models are also predictive, and we expect that future work will tune the model to optimize network recovery using AUPR, or other model selection appropriate metrics, of held out gold standard network edges, by maximizing R^2 for predictive power.

SupirFactor provides a novel metric for evaluating DNN architectures, specifically designed for the needs of GRN inference. Explained relative variance (ERV) estimates the importance of each latent feature to each output feature, directly and indirectly. This metric has appealing properties; ERV is bounded and facilitates ranking regulatory relationships and discarding non-predictive model weights. DNN models are often overparameterized, which results in problems interpreting model weights for GRN inference. As biological GRNs are sparse model selection must be part of the evaluation of any GRN inference algorithm. The model evaluation metric and model selection criteria we propose are also useful for evaluating the contribution of intermediate DNN layers which are not explicitly defined as TFs. By using ERV to evaluate linkages from regulatory TFs, through a latent meta-TF layer, to target genes, we are able to use the meta-TF layer as a powerful pathway analysis tool.

We demonstrate the validity of ERV by comparing it directly to GRN inference using model weights alone, and find that it improves GRN inference link interpertation. ERV also allows for post-training analysis of any gene expression data to determine which parts of the network are specific to that context. We show this by extracting context-specific networks from the *S. cerevisiae* single cell data set, which contains observations from fifteen different growth conditions. The recovery of these post-training context-specific networks is an improvement over previous work, which requires that the context is embedded into the model pre-training. SupirFactor is therefore a valuable tool to identify context specific and contextual regulatory interactions.

The driving features of GRNs can be condensed into TFs. Another core concept explored in this work is that of latent feature inference and interpertation, *i.e.* TF activity (TFA). In the model organisms *S. cerevisiae* we demonstrate that the model latent feature activity, the TFA of a TF is distinct from the expression of that same TF. We do this by studying the cell-cycle where we see a clear delay of relevant TFA compared to expression. Demonstrating that reliance on expression of TF as independant features to model a GRN does not capture the regulatory structure of GRNs. A drawback in many works related to GRN inference.

To demonstrate that SupirFactor scales to the complexity of mammalian systems we evaluate a model learned from a PBMC multi-ome single cell dataset, and characterize the pattern of TFA and functional enrichment in different contexts. The model make use of context specific prior evidence to further restrict TF variance. And we find that we can extract functional enrichment based on annotated celltypes reliably.

Reuse of computational models can be valuable as a tool to understand and conceptualize new experimental data evident by recent reuse of single cell sequencing atlases in the field of genomics[48, 58]. Unfortunately, the reuse of GRNs themselves is rare, and for most studies gene regulatory networks are inferred entirely based on new data. We consider this to be a general limitation in current-generation GRN inference models, which do not have mechanisms to embed new data into an existing GRN. SupirFactor tackles this by using a DNN architecture together with the transparency framework (ERV). We demonstrate this reuse by embedding novel data and by contextually analysing sub-networks after the model has been trained, gaining insight not explicitly provided to the model before training.

Developing explainable deep learning models for GRN inference is a critical requirement for improving models of gene expression and regulation[23]. The goal of this work was to build a formalized GRN inference model with explicit optimization and objective functions, from which latent states can be directly interpreted. The resulting formalism, SupirFactor, is a powerful GRN inference tool with additional pathway analysis and protein activity functionality, that can be applied to both bulk and single cell data. SupirFactor can harmonise regulatory evidence, epigenetic data and expression readout in a regulatory and functionally meaningful way. While challenges still exist, like model stability and model selection; tightly connected to the nature of non-linear machine learning algorithms, advances in single cell multi-omics and epigenetic sequencing are steadfast and will further narrow the specificity in model constraints with its inclusions.

With additional work related to architecture, algorithm development, and prior evidence construction, the framework can be further extended and prove even more useful.

4 Methods

4.1 *StrUcture Primed Inference of Regulation using latent Factor ACTivity* (SupirFactor)

We define two SupirFactor models. The shallow model, used mainly for testing, which consists of a single layer that represents individual TFs and their activity. The hierarchical model, which consists of two layers, the first representing individual TFs and their activity, and the second representing TFs aggregated into pathways, termed meta TFs (mTFs). The hierarchical model is the main model used in this work.

4.1.1 Shallow SupirFactor

Starting from our model framework (Section 2.1), gene expression is a function of TFA and is used as the independent feature to weight influence on gene expression from TFs. We can formulate the problem as

$$\hat{\boldsymbol{x}} = h(\boldsymbol{x}) = (g \circ f)(\boldsymbol{x}) \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the gene expression of one observation with n genes, with g, f as functions of the form

$$\boldsymbol{\phi} = f(\boldsymbol{x}) = \sigma\left(\boldsymbol{W}\boldsymbol{x}\right) \tag{2}$$

and

$$\hat{\boldsymbol{x}} = g(\boldsymbol{\phi}) = \sigma\left(\boldsymbol{\Theta}\boldsymbol{\phi}\right) \tag{3}$$

with the activation function σ, aggregating the linear combination of inputs for each latent feature. The linear combination of inputs without activation is similar to the NCA framework described in section 4.4.4 with the distinction that this formulation weighs $\boldsymbol{W}$ instead of fitting $\boldsymbol{\phi}$ with a static $\boldsymbol{P}$.

$\boldsymbol{\Theta}$ and $\boldsymbol{W}$ are the weighted connections of input features to output features in a single layer, with equation (3) corresponding to the shallow SupirFactor (Figure 1C). $\boldsymbol{\phi}$ is the inferred latent activity interpreted as TFA of expression mapped through f. We set $\boldsymbol{W} \in \boldsymbol{P}$, such that the sparsity structure of $\boldsymbol{W}$ is identical to $\boldsymbol{P}$. This ensures that (I) $\boldsymbol{\phi} \in \mathbb{R}^K$, with K TFs, constraining the informative genes influences to those in the prior $\boldsymbol{P}$, and (II) the causal flow from regulator to target defined by the prior is enforced. Causality does not imply direct binding but rather in this case the variance of each TF being constrained to the variance of its targets as opposed to the covariance of the TF expression.

4.1.2 Hierarchical SupirFactor

In Hierarchical SupirFactor (Figure 1A) we add an additional mTF layer between TFs and output gene expression. This allows higher order interactions between TFs (representing biological concepts like redundancy, competition, and physical complexing), and other conditional non-linear dependencies to be modelled. This extends the formulation of SupirFactor so that it can generate TF interaction hypotheses and be used as a tool for pathway analysis.

$$\hat{\boldsymbol{x}} = h(\boldsymbol{x}) = (g \circ s \circ f)(\boldsymbol{x}) \tag{4}$$

where

$$\hat{\boldsymbol{\phi}} = s(\boldsymbol{\phi}) = \sigma\left(\boldsymbol{\Pi}\boldsymbol{\phi}\right) \tag{5}$$

and

$$\hat{\boldsymbol{x}} = g(\hat{\boldsymbol{\phi}}) = \sigma\left(\boldsymbol{\Theta}\hat{\boldsymbol{\phi}}\right) \tag{6}$$

$\boldsymbol{\Pi}$ is the weight matrix of TF-TF interactions that maps individual TF activity to the mTF layer.

4.2 Joint representation learning for context specific constraints

Joint representation learning is a transfer learning method where context-specific evidence is aggregated into a common model structure [see 44, chap. 15] (Figure 1E). This is implemented in SupirFactor by adding a biological context-specific constraint on the prior evidence. We define $\boldsymbol{P}_C$ as prior evidence for C where C is a biological context, like a cell type, growth condition, or temporal group. Weights $\boldsymbol{W}_C \in \boldsymbol{P}_C$ are also context-specific, and are mapped jointly through $\boldsymbol{\Pi}$ and $\boldsymbol{\Theta}$ that are common to all contexts. Experimental data is labeled with the appropriate context and data for each context is submitted batch-wise to the model for training. Context weights $\boldsymbol{W}_C$ are individually trained and may vary between contexts if $\boldsymbol{P}_C$ is the same. Equation (4) then takes the form;

$$\hat{\boldsymbol{x}}_C = (g \circ s \circ f_C)(\boldsymbol{x}_C) \tag{7}$$

where f_C is the context specific structure $(\boldsymbol{P}_C)$ primed encoder.

4.2.1 Fitting model

To train the SupirFactor model, $\boldsymbol{W}$, $\boldsymbol{\Pi}$, and $\boldsymbol{\Theta}$ are fit to minimize mean squared error between $\hat{\boldsymbol{x}}$ and $\boldsymbol{x}$.

$$\text{MSE} = \frac{1}{N} \sum_i \|\boldsymbol{x}_i - h(\boldsymbol{x}_i)\|^2 \tag{8}$$

This is implemented as minimization by batch stochastic gradient descent with the Adam solver [59] and pytorch [60].

Nonzero encoder weights are initially mirrored and made non-negative in both the decoder $\boldsymbol{\Theta}$ and encoder $\boldsymbol{W}$, although all elements in $\boldsymbol{\Theta}$ are free to be fit by the solver. The prior assumption is that TFA is positively correlated with gene expression during model initialisation if no other evidence is available.

4.3 Model regularization

4.3.1 Parameter penalties

Model weights are penalized, regularizing models to mitigate overfitting and to balance bias and variance [61]. We use a *weight decay* factor, corresponding to a ridge penalty [see 62, section. 6.7.6]. The objective function to be minimized is then

$$\text{MSE} + \zeta \|\boldsymbol{W}\|_2 + \zeta \|\boldsymbol{\Theta}\|_2 + \zeta \|\boldsymbol{\Pi}\|_2 \tag{9}$$

where ζ is the ridge penalty applied. The ζ parameter is set by cross validation, splitting the data into equal training and validation sets and evaluating model performance where $\zeta \in \{0, 10^{-10}, 10^{-9}, \ldots, 10^{-1}\}$ [33, 63].

4.3.2 Dropout

Dropout is an additional regularization method [64] where a fraction of nodes are removed from each sample during training, mitigating the risk that noise in the data will trap the model in a local minimum. Dropout can be applied to input or to latent layers. In short, training data is randomly batched into groups, and each batch is then used to train a network where a fraction, p of data points are removed randomly from each sample in each batch before feeding it through selected layer(s). This is implemented in SupirFactor through the Dropout module in pytorch. Droput is tested by cross validation as above on both the input and the latent TF layer, searching $p \in \{0, 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$.

4.4 Explained Relative Variance (ERV)

4.4.1 Model selection and model parameter ranking

GRNs are sparse and most genes have a limited number of directly regulating TFs. The importance of model parameters is quantified, and relatively unimportant parameters are shrunk to zero. Two different ways of ranking inferred interactions are evaluated in this work: (I), Ranking the magnitude of model weights $|\theta_{i,k}|$ (MODEL), and (II), ranking interactions by their explained relative error (ERV). ERV perturbs latent features and quantifies the consequence of that perturbation [37]. The goal is not to eliminate redundancy, but rather to eliminate over-parameterization and constrain the parameter space of SupirFactor.

ERV is calculated as coefficient of partial determination [43] so the bound on the error contributing to predict gene expression can be evaluated $\xi^2 \in \,]-\infty, 1]$ where a predictive link has an $\xi^2 \in \,]0, 1]$. The GRN model is trained once, as re-training for each perturbed latent feature is computationally intractable. ERV is determined from the ratio of the full model MSE to the perturbed model MSE.

$${\xi^2}_{i,k} = 1 - \frac{\text{MSE}_i\left(\bar{h}(\boldsymbol{x})_{\boldsymbol{\phi}}\right)}{\text{MSE}_i\left(\bar{h}(\boldsymbol{x})_{\boldsymbol{\phi}_{\mathrm{K} \neq k}}\right)} \tag{10}$$

$\bar{h}$ is the trained model, $\text{MSE}_i\left(\bar{h}(\boldsymbol{x})_{\boldsymbol{\phi}}\right)$ is the MSE of the full, unperturbed model for gene i, and $\text{MSE}_i\left(\bar{h}(\boldsymbol{x})_{\boldsymbol{\phi}_{\text{K}\neq k}}\right)$ is the MSE of the perturbed model for gene i where the activation of the latent layer for regulator k is set to 0. Model parameters for each gene i are ranked by the value of ${\xi^2}_{i,k}$ for all k.

4.4.2 Ranking model parameters for models with multiple layers

To be able to interpret any parameter in the model we use the ERV concept (section 4.4.1). To determine the importance in any hidden layer L not directly connected to the output we compute two ξ^2 matrices. ${\xi^2}_L$ and ${\xi^2}_{L+1}$. Where L indicates the layer with corresponding latent feature input to the layer in question and $L+1$ the next layer with corresponding latent feature input. For layer weights $\mathbf{\Pi}$ we compute each element $\pi_{m,k}$ with m as output feature and k as input feature by first computing the vectors ${\xi^2}_{:,k,L}$ and ${\xi^2}_{:,m,L+1}$. That is, the ξ^2 of each element of $\mathbf{\Theta}_{:,m}$ and the ξ^2 of each element of the indirect contribution of latent feature k in layer L to the output genes ${\xi^2}_{:,k,L}$. To eliminate weights in $\mathbf{\Pi}$ we threshold ${\boldsymbol{\xi}^{\mathbf{2}}}_{:,k,L+1} > \epsilon_1$ and ${\boldsymbol{\xi}^{\mathbf{2}}}_{:,m,L} \;> \epsilon_2$ and compute the ERV;

$$
{\xi^2}_{m,k,\mathbf{\Pi}} \equiv 1 - \frac{\sum_{i \in k \cap m} \text{MSE}_i\left(\bar{h}(\boldsymbol{x})_{\boldsymbol{\phi}}\right)}{\sum_{i \in k \cap m} \text{MSE}_{i,k,L}\left(\bar{h}(\boldsymbol{x})_{\boldsymbol{\phi}_{\text{K}\neq k}}\right)} \tag{11}
$$

where $i \in k \cap m$ is the intersection set of predictive TF to gene interactions for latent feature k from layer L, and m for layer $L+1$. If $k \cap m = \emptyset$ then ${\xi^2}_{m,k,\mathbf{\Pi}} \equiv 0$.

The classical GRN realisation is interpreted as the indirect connections from ${\xi^2}_{i,k,L}$, connecting the latent input features in layer L, k to the output target genes i. With L in this case representing the TFA activation layer.

4.4.3 Stable architecture and non predictive weight elimination

To select an interpretable model we want to reduce the model size in terms of individual weights to arrive at a model with parameters that are predictive and stable. We define predictive as when an individual parameter that can be determined to connect an upstream regulator to its downstream target, has $\xi^2 > \epsilon$. Stable in this case means that the parameter is predictive on unseen data when the model is trained on reduced subset of parameters.

We apply model constraints on individual weights. Parameter weights $\hat{\theta}_{i,k}$ (and $\theta_{i,m}$ in hierarchical SupirFactor), with i output and k, and m input features are removed if ${\xi^2}_{i,k,L} \leq \epsilon$ for the respective layer L, with a choice of $\epsilon = 0$ *i.e.* not predictive of the output, is the most conservative.

To enforce these constraints in number of regulators per gene we use a model selection step after an initial training run (Figure 1E). The model selection step is derived from ξ^2 where

$$
\widehat{\xi^2}_{i,k} \equiv \frac{{\xi^2}_{i,k}}{\overline{\xi^2}_i} \tag{12}
$$

with $\overline{\xi^2}_i$ as the maximum ξ^2 for gene i. Selection is done iteratively by selecting a threshold ϵ. The model is then refit with parameters where $\widehat{\xi^2}_{i,k} > \epsilon$. This is done iteratively until convergence where at each step ξ^2 is recomputed. ϵ is a measure of inclusion of relative predictive power. Using $\epsilon = 0$ means all predictive links are kept after sub-setting and relative predictive power $\widehat{\xi^2}$ is not impacting the subset.

Using $\widehat{\xi^2}_{i,k}$ facilitate selecting regulator subsets where, unless the gene is too noisy or the prior lack sufficient information that can reliably predict the specific expression pattern (*i.e.* all ${\xi^2}_k \leq \epsilon$ for the gene in question), at least 1 TF can be inferred to be predictive relative to all k for that gene i and other regulators are ranked relative to it.

For the hierarchical model sub-setting, weights are eliminated if no predictive interactions can be derived from the indirect path between a latent feature k through the latent feature m in the subsequent layer to the set of joint output features, above the chosen ϵ threshold. If ${\xi^2}_{k,m,\mathbf{\Pi}} \leq 0$ the hidden layer weight is pruned.

4.4.4 Network Component Analysis (NCA)

Network component analysis (NCA) can be used to estimate TFA directly by formulating the causal network inference problem so that

$$\boldsymbol{Y} = \boldsymbol{\Phi}\boldsymbol{\Theta} \tag{13}$$

where $\boldsymbol{Y} \in \mathbb{R}^{S,n}$ is gene expression with n genes and S samples, $\boldsymbol{\Phi} \in \mathbb{R}^{m,k}$ is the TFA with k TFs, and $\boldsymbol{\Theta} \in \mathbb{R}^{k,n}$ is the regulatory effects linking genes to TFs. The unknown "true" $\boldsymbol{\Theta}$ is the regulatory interaction between genes and TFs we want to find. $\boldsymbol{\Phi}$ is unknown given the assumption that the expression level of a transcription factor k does not correlate well with the activity of the protein [65]. Therefore we need to solve for both $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$, which forces us to convolve our estimation of regulatory effect and the TFA. To deconvolve and solve this we impose a prior $\boldsymbol{P}$ with elements $\in \{0, 1\}$ as an initial guess to the structure of $\boldsymbol{\Theta}$ and use that to solve for an initial estimate for $\boldsymbol{\Phi}$.

$$\hat{\boldsymbol{\Phi}} = \arg\min_{\boldsymbol{\Phi}} \|\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{P}\| \tag{14}$$

This is solved by ordinary least squares. The estimated $\boldsymbol{\Phi}$ is then used to solve for $\boldsymbol{\Theta}$

$$\hat{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}} \left\|\boldsymbol{Y} - \hat{\boldsymbol{\Phi}}\boldsymbol{\Theta}\right\| \tag{15}$$

This is interpreted as an estimate of TFA given a number of reporter genes defined by the prior $\boldsymbol{P}$, *i.e.* the expression level of the target genes is a proxy for how active a TF is in any given sample. The variance of the TF activity is defined and constrained by the variance of the reporter genes.

4.5 Epigenetic masking

To incorporate the paired ATAC- and RNA- sequencing data we create a masking scheme to mask input gene expression profiles with the ATAC so that expression input

$$\dot{x}_{i,s} = x_{i,s} \cdot 1 \text{ if } a_{p,s} \in A_i \text{ else } 0 \tag{16}$$

with $x_{i,s}$ being the expression of gene i in sample s and $a_{p,s}$ being an available ATAC peak p in the set of accessible regions A_i associated with gene i.

4.6 Data Acquisition

4.6.1 Bulk expression data

Bacillus subtilis bulk expression data for data set 1 (B1) [10] and data set 2 (B2) [66], and the known prior network [10] were used as previously described [41]. *Saccharomyces cerevisiae* bulk expression data for data set 1 [67] and data set 2 [68], and the known prior network [69] were used as previously described [41].

4.6.2 Single cell expression data

Saccharomyces cerevisiae single cell training data was assembled from [13] and [70] as previously described [41].

S. cerevisiae test data was collected using a method previously published [13]. In short, biological replicates containing unique, transcriptionally-expressed molecular barcodes of a wild-type strain 1 (*MATα/MATa Δho::NatMX/Δho::KanMX*) and a wild-type strain 2 (*MATα/MATa HO/Δho::NatMX HAP1+::pACT1-Z3EV::NatMX/HAP1 ura3Δ0/URA3 can1Δ::prSTE2-HIS5/CAN1 HIS3/his3Δ1 LYP1/lyp1Δ0*) were generated as previously described [13].

Strains were grown overnight in rich media (YPD as previously described [13]) and then subcultured into 100mL YPD or minimal media (MMD as previously described [13]) for 3 hours. Cells from each flask were then taken, fixed with saturated ammonium sulfate, processed, and sequenced using the protocol as previously described [13]. Raw sequencing data was processed into count data using a previously-described pipeline [13] which joined the transcriptional barcodes to individual cells, assigning specific genotypes to cells and removing any cell containing multiple distinct barcodes as doublets. Four data sets were then created from this count table; YPD 1 (n=1531, wild-type strain 1 in rich YPD media), YPD 2 (n=1428, wild-type strain 2 in rich YPD media), MMD 1 (n=492, wild-type strain 1 in minimal MMD media), and MMD 2 (n=463, wild-type strain 2 in minimal MMD media). This data is deposited in NCBI GEO as GSE218089.

4.6.3 Yeast Annotations

Cell cycle related yeast genes are annotated based on [71]. Ribosomal, ribosomal biogenesis, and induced environmental stress response genes are annotated based on [72].

Individual cells in single cell RNA expression data sets are assigned a cell cycle phase based on cell cycle gene annotations. Expression of each cell cycle gene is normalized to a mean of 0 and unit variance. All marker genes annotated with a specific cell cycle phase (G1, S, G2, M, or M/G1) are grouped, and the cell is assigned to the phase that has the maximum mean group expression.

Kyoto Encyclopedia of Genes and Genomes (KEGG) annotations [47] were selected to cover the majority of the core yeast metabolism. KEGG annotations are KEGG:04111 (Cell cycle), KEGG:00010 (Glycolysis), KEGG:00020 (TCA cycle), KEGG:00500 (Starches), KEGG:00660 (C5-Branched), KEGG:01210 (C5-Branched), KEGG:00250 (AA (A, D, E, N, Q)), KEGG:00260 (AA (G, S, T)), KEGG:00330 (AA (R & P)), KEGG:00400 (AA (F, Y, W)), KEGG:00270 (AA (C & M)), KEGG:00920 (Sulfur), KEGG:00061 (Fatty acids), KEGG:00230 (Purines), and KEGG:00240 (Pyrimidines)

4.6.4 PBMC multi-ome dataset prepocessing

Paired PBMC scRNA-seq and scATAC-seq was downloaded from the 10x website (https://support.10xgenomics.com/single-cell-multiome-atac-gex/datasets/1.0.0/pbmc_granulocyte_sorted_10k). This data was preprocessed using a previously published workflow [48]. In short, the RNA-seq data is preprocessed as detailed in Section 4.7.1, with the additional filtering of cells with > 25000 or < 1000 counts and $< 20\%$ mitocondrial counts of total. For the ATACseq data we used epiScanpy[73], filtereing peaks in < 10 cells and cells with < 5000 or $> 7 \cdot 10^4$ counts, and with a variability score < 0.515. Final data contains 10411 cells, 21601 genes and 75111 peaks,

Cell types are annotated using the reference PBMC dataset[58] passed to scanpy's [74] inject label transfer function, resulting in 8 annotated celltypes (Figure 5A).

4.6.5 ENCODE PBMC prior knowledge network construction

TF-ChIP peaks were obtained as narrowPeak BED files from the ENCODE project database. The GRCh38 genomic annotations (NCBI GCF_000001405.39) were obtained as a GTF file from NCBI and filtered for protein-coding genes.

TF-ChIP peaks were linked to candidate target genes with the inferelator-prior package [41]. TF peaks were annotated as possible regulators of a gene if they were within 50kbp upstream of a gene transcription start site and 2kbp downstream of a gene transcription site, with no other gene between the TF peak and the gene transcription start site. TF Peaks were further filtered to remove any regulators not annotated as TFs (including GTFs, chromatin modifiers, and polymerase subunits).

This large pool of potential TF regulatory peaks were then subset for intersetion with annotated regulatory regions for PBMC cell types (ENCODE Accession IDs ENCFF776AJJ, ENCFF497NXM, ENCFF984SPH, ENCFF079TQT, ENCFF862ULW, ENCFF504FDC, and ENCFF905BHJ). The peak intensity (signalValue) was summed for all peaks annotated to each TF-gene pair to generate a genes by TFs putative regulatory matrix. This matrix was further constrained for sparsity by retaining at most 1.5% non-zero values for each TF, shrinking all values below this threshold to zero, and producing a genes by TFs prior knowledge network matrix with a sparsity of 1.22%.

4.7 Data preprocessing and model parameterisation

4.7.1 Single cell pre-processing

For the single cell data, unless otherwise stated, we follow standard normalization procedures which include, (i) filtering genes with expression in < 10 cells, (ii) count normalization; scaling each cells total count to the same value over the dataset. This serves to eliminate the effect of variable sequencing depth in the experimental technique, and (iii) log transforming the (data + 1) using the natural logarithm.

4.7.2 Feature normalization

For bulk data we use the standard normalization of each input feature so that

$$z = \frac{x - \mu}{\sigma}$$

for each gene with μ = mean and σ = standard deviation over the gene.

To preserve the sparse structure of single cell data we, in addition to the above, adopt a robust normalization approach without centering. Each gene is scaled by the range of the 1 and 99 percentile and shifted so the lowest value for each gene = 0 implemented using the scikit-learn RobustScaler method [75] which we call RobustMinScaler.

4.8 ReLU in hierarchical SupirFactor

For DNN linear activation does not contribute meaningfully in different layers and can be reduced to a single linear map. The rectified linear unit (ReLU)[44] truncates activation to stay strictly positive and injects non-linearities into the model architecture. For hierarchical SupirFactor we therefore use ReLU and define the gradient for the ReLU function

$$\mathrm{ReLU}_0 \equiv \sigma(z) = \max(0, z)$$

so that with $z = 0$ the gradient is $\equiv 1$. With z as the linear combination of inputs to each feature.

4.9 Visualisation

Visualisations throughout this work, if not stated otherwise, was generated in Matplotlib[76] with some components done with seaborn[77] and scanpy[74].

5 Acknowledgement

We thank past and present members of the Christiaen, Bonneau, and Gresham labs for discussions and valuable feedback on this manuscript. This work was supported in part through the NYU's IT High Performance Computing resources, services, and staff expertise.

6 Funding

This work was supported by the NIH (R01HD096770, RM1HG011014 R01NS116350, R01NS118183, R01GM134066, R01GM107466), and the Simons Foundation.

References

[1] Deborah Chasman, Alireza Fotuhi Siahpirani, and Sushmita Roy. Network-based approaches for analysis of complex biological systems. *Current Opinion in Biotechnology*, 39:157–166, 2016. ISSN 0958-1669. doi: https://doi.org/10.1016/j.copbio.2016.04.007. URL https://www.sciencedirect.com/science/article/pii/S095816691630115X. Systems biology, Nanobiotechnology.

[2] Kitt D. Paraiso, Jin S. Cho, Junseok Yong, and Ken W.Y. Cho. Chapter two - early xenopus gene regulatory programs, chromatin states, and the role of maternal transcription factors. In Isabelle S. Peter, editor, *Gene Regulatory Networks*, volume 139 of *Current Topics in Developmental Biology*, pages 35 – 60. Academic Press, 2020. doi: https://doi.org/10.1016/bs.ctdb.2020.02.009. URL http://www.sciencedirect.com/science/article/pii/S0070215320300351.

[3] Dong-Yeon Cho, Yoo-Ah Kim, and Teresa M. Przytycka. Chapter 5: Network biology approach to complex diseases. *PLOS Computational Biology*, 8(12):1–11, 12 2012. doi: 10.1371/journal.pcbi.1002820. URL https://doi.org/10.1371/journal.pcbi.1002820.

[4] Marylyn D. Ritchie, Emily R. Holzinger, Ruowang Li, Sarah A. Pendergrass, and Dokyoon Kim. Methods of integrating data to uncover genotype–phenotype interactions. *Nature Reviews Genetics*, 16(2): 85–97, Feb 2015. ISSN 1471-0064. doi: 10.1038/nrg3868. URL https://doi.org/10.1038/nrg3868.

[5] Debbie K. Goode, Nadine Obier, M.S. Vijayabaskar, Michael Lie-A-Ling, Andrew J. Lilly, Rebecca Hannah, Monika Lichtinger, Kiran Batta, Magdalena Florkowska, Rahima Patel, Mairi Challinor, Kirstie Wallace, Jane Gilmour, Salam A. Assi, Pierre Cauchy, Maarten Hoogenkamp, David R. Westhead, Georges Lacaud, Valerie Kouskoff, Berthold Göttgens, and Constanze Bonifer. Dynamic gene regulatory networks drive hematopoietic specification and differentiation. *Developmental Cell*, 36(5): 572–587, 2016. ISSN 1534-5807. doi: https://doi.org/10.1016/j.devcel.2016.01.024. URL https://www.sciencedirect.com/science/article/pii/S1534580716300028.

[6] Yelena Bernadskaya and Lionel Christiaen. Transcriptional control of developmental cell behaviors. *Annual Review of Cell and Developmental Biology*, 32(1):77–101, 2016. doi: 10.1146/annurev-cellbio-111315-125218. URL https://doi.org/10.1146/annurev-cellbio-111315-125218. PMID: 27501446.

[7] DS Latchman. Transcription factors: an overview. *International journal of experimental pathology*, 74 (5):417—422, October 1993. ISSN 0959-9673. URL https://europepmc.org/articles/PMC2002184.

[8] Antonis Papachristodoulou and Ben Recht. Determining interconnections in chemical reaction networks. *2007 American Control Conference*, pages 4872–4877, July 2007. ISSN 0743-1619. doi: 10.1109/ACC.2007.4283084. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4283084.

[9] Theresa Schacht, Marcus Oswald, Roland Eils, Stefan B. Eichmüller, and Rainer König. Estimating the activity of transcription factors by the effect on their target genes. *Bioinformatics*, 30(17):i401–i407, 08 2014. ISSN 1367-4803. doi: 10.1093/bioinformatics/btu446. URL https://doi.org/10.1093/bioinformatics/btu446.

[10] Mario L Arrieta-Ortiz, Christoph Hafemeister, Ashley Rose Bate, Timothy Chu, Alex Greenfield, Bentley Shuster, Samantha N Barry, Matthew Gallitto, Brian Liu, Thadeous Kacmarczyk, Francis Santoriello, Jie Chen, Christopher DA Rodrigues, Tsutomu Sato, David Z Rudner, Adam Driks, Richard Bonneau, and Patrick Eichenberger. An experimentally supported model of the bacillus subtilis global transcriptional regulatory network. *Molecular Systems Biology*, 11(11), 2015. ISSN 1744-4292. doi: 10.15252/msb.20156236. URL http://msb.embopress.org/content/11/11/839.

[11] Qianqian Shi, Chuanchao Zhang, Weifeng Guo, Tao Zeng, Lina Lu, Zhonglin Jiang, Ziming Wang, Juan Liu, and Luonan Chen. Local network component analysis for quantifying transcription factor activities. *Methods*, 124:25 – 35, 2017. ISSN 1046-2023. doi: 10.1016/j.ymeth.2017.06.018. URL http://www.sciencedirect.com/science/article/pii/S1046202317300506. Integrative Analysis of Omics Data.

[12] Katy C. Kao, Young-Lyeol Yang, Riccardo Boscolo, Chiara Sabatti, Vwani Roychowdhury, and James C. Liao. Transcriptome-based determination of multiple transcription regulator activities in escherichia coli by using network component analysis. *Proceedings of the National Academy of Sciences of the United States of America*, 101(2):641–646, 2004. doi: 10.1073/pnas.0305287101. URL http://www.pnas.org/content/101/2/641.abstract.

[13] Christopher A Jackson, Dayanne M Castro, Giuseppe-Antonio Saldi, Richard Bonneau, and David Gresham. Gene regulatory network reconstruction using single-cell rna sequencing of barcoded genotypes

in diverse environments. *eLife*, 9:e51254, jan 2020. ISSN 2050-084X. doi: 10.7554/eLife.51254. URL https://doi.org/10.7554/eLife.51254.

[14] Dayanne M. Castro, Nicholas R. de Veaux, Emily R. Miraldi, and Richard Bonneau. Multi-study inference of regulatory networks for more accurate models of gene regulation. *PLOS Computational Biology*, 15(1):1–22, 01 2019. doi: 10.1371/journal.pcbi.1006591. URL https://doi.org/10.1371/journal.pcbi.1006591.

[15] Kenji Kamimoto, Christy M. Hoffmann, and Samantha A. Morris. Celloracle: Dissecting cell identity via network inference and in silico gene perturbation. *bioRxiv*, 2020. doi: 10.1101/2020.02.17.947416. URL https://www.biorxiv.org/content/early/2020/04/21/2020.02.17.947416.

[16] Yifei Chen, Yi Li, Rajiv Narayan, Aravind Subramanian, and Xiaohui Xie. Gene expression inference with deep learning. *Bioinformatics*, 32(12):1832–1839, 02 2016. ISSN 1367-4803. doi: 10.1093/bioinformatics/btw074. URL https://doi.org/10.1093/bioinformatics/btw074.

[17] Lujia Chen, Chunhui Cai, Vicky Chen, and Xinghua Lu. Learning a hierarchical representation of the yeast transcriptomic machinery using an autoencoder model. *BMC Bioinformatics*, 17(1):S9, 2016. ISSN 1471-2105. doi: 10.1186/s12859-015-0852-1. URL https://doi.org/10.1186/s12859-015-0852-1.

[18] Davide Chicco, Peter Sadowski, and Pierre Baldi. Deep autoencoder neural networks for gene ontology annotation predictions. In *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, BCB '14, page 533–540, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450328944. doi: 10.1145/2649387.2649442. URL https://doi.org/10.1145/2649387.2649442.

[19] Sanjiv K. Dwivedi, Andreas Tjärnberg, Jesper Tegnér, and Mika Gustafsson. Deriving disease modules from the compressed transcriptional space embedded in a deep autoencoder. *Nature Communications*, 11(1):856, 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-14666-6. URL https://doi.org/10.1038/s41467-020-14666-6.

[20] Hung-I Harry Chen, Yu-Chiao Chiu, Tinghe Zhang, Songyao Zhang, Yufei Huang, and Yidong Chen. Gsae: an autoencoder with embedded gene-set nodes for genomics functional characterization. *BMC Systems Biology*, 12(8):142, 2018. ISSN 1752-0509. doi: 10.1186/s12918-018-0642-2. URL https://doi.org/10.1186/s12918-018-0642-2.

[21] Ye Yuan and Ziv Bar-Joseph. Deep learning for inferring gene relationships from single-cell expression data. *Proceedings of the National Academy of Sciences*, 116(52):27151–27158, 2019. ISSN 0027-8424. doi: 10.1073/pnas.1911536116. URL https://www.pnas.org/content/116/52/27151.

[22] Romain Lopez, Jeffrey Regier, Michael B. Cole, Michael I. Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15(12):1053–1058, Dec 2018. ISSN 1548-7105. doi: 10.1038/s41592-018-0229-2. URL https://doi.org/10.1038/s41592-018-0229-2.

[23] Gherman Novakovsky, Nick Dexter, Maxwell W. Libbrecht, Wyeth W. Wasserman, and Sara Mostafavi. Obtaining genetics insights from deep learning via explainable artificial intelligence. *Nature Reviews Genetics*, Oct 2022. ISSN 1471-0064. doi: 10.1038/s41576-022-00532-2. URL https://doi.org/10.1038/s41576-022-00532-2.

[24] Ian Covert, Scott Lundberg, and Su-In Lee. Explaining by removing: A unified framework for model explanation, 2020. URL https://arxiv.org/abs/2011.14878.

[25] A.H. Sung. Ranking importance of input parameters of neural networks. *Expert Systems with Applications*, 15(3):405–411, 1998. ISSN 0957-4174. doi: https://doi.org/10.1016/S0957-4174(98)00041-4. URL https://www.sciencedirect.com/science/article/pii/S0957417498000414.

[26] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, Oct 2019. ISSN 1573-1405. doi: 10.1007/s11263-019-01228-7. URL http://dx.doi.org/10.1007/s11263-019-01228-7.

[27] Nikolaus Fortelny and Christoph Bock. Knowledge-primed neural networks enable biologically interpretable deep learning on single-cell sequencing data. *Genome Biology*, 21(1):190, 2020. ISSN 1474-760X. doi: 10.1186/s13059-020-02100-5. URL https://doi.org/10.1186/s13059-020-02100-5.

[28] Uri Alon. *An introduction to systems biology: design principles of biological circuits*, volume 10 of *Chapman & Hall/CRC mathematical and computational biology series*. Chapman & Hall/CRC, 1 edition, 2007. ISBN 1584886420. URL http://www.loc.gov/catdir/enhancements/fy0654/2005056902-d.html.

[29] D Di Bernardo, T S Gardner, and J J Collins. Robust identification of large genetic networks. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 497:486–497, January 2004. ISSN 1793-5091. URL http://www.ncbi.nlm.nih.gov/pubmed/14992527.

[30] Richard Bonneau, David J Reiss, Paul Shannon, Marc Facciotti, Leroy Hood, Nitin S Baliga, and Vesteinn Thorsson. The inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome biology*, 7(5):R36, January 2006. ISSN 1465-6914. doi: 10.1186/gb-2006-7-5-r36. URL http://genomebiology.com/2006/7/5/R36.

[31] Richard Bonneau. Learning biological networks: from modules to dynamics. *Nature chemical biology*, 4 (11):658–64, November 2008. ISSN 1552-4469. doi: 10.1038/nchembio.122. URL http://dx.doi.org/10.1038/nchembio.122http://www.ncbi.nlm.nih.gov/pubmed/18936750.

[32] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction.* Springer, 2 edition, 2009. URL http://www-stat.stanford.edu/~tibs/ElemStatLearn/.

[33] Andreas Tjärnberg, Torbjörn E. M. Nordling, Matthew Studham, and Erik L. L. Sonnhammer. Optimal sparsity criteria for network inference. *Journal of Computational Biology*, 20(5):398–408, May 2013. ISSN 1066-5277. doi: 10.1089/cmb.2012.0268. URL http://dx.doi.org/10.1089/cmb.2012.0268.

[34] Stephanie Kamgnia Wonkap and Gregory Butler. Benin: Biologically enhanced network inference. *Journal of Bioinformatics and Computational Biology*, 18(03):2040007, 2020. doi: 10.1142/S0219720020400077. URL https://doi.org/10.1142/S0219720020400077. PMID: 32698722.

[35] Aviv Madar, Alex Greenfield, Eric Vanden-Eijnden, and Richard Bonneau. Dream3: network inference using dynamic context likelihood of relatedness and the inferelator. *PloS one*, 5(3):e9803–e9803, Mar 2010. ISSN 1932-6203. doi: 10.1371/journal.pone.0009803. URL https://pubmed.ncbi.nlm.nih.gov/20339551.

[36] Vân Anh Huynh-Thu, Alexandre Irrthum, Louis Wehenkel, and Pierre Geurts. Inferring regulatory networks from expression data using tree-based methods. *PLoS ONE*, 5(9):1–10, 09 2010. doi: 10.1371/journal.pone.0012776. URL http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0012776.

[37] Rasmus Magnusson and Mika Gustafsson. LiPLike: towards gene regulatory network predictions of high certainty. *Bioinformatics*, 36(8):2522–2529, 01 2020. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz950. URL https://doi.org/10.1093/bioinformatics/btz950.

[38] C. Bauckhage, C. Ojeda, J. Schücker, R. Sifa, and S. Wrobel. Informed machine learning through functional composition. In *LWDA*, 2018.

[39] Jason D. Buenrostro, Paul G. Giresi, Lisa C. Zaba, Howard Y. Chang, and William J. Greenleaf. Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, dna-binding proteins and nucleosome position. *Nature Methods*, 10:1213 EP –, Oct 2013. URL http://dx.doi.org/10.1038/nmeth.2688. Article.

[40] Tong Ihn Lee, Nicola J. Rinaldi, François Robert, Duncan T. Odom, Ziv Bar-Joseph, Georg K. Gerber, Nancy M. Hannett, Christopher T. Harbison, Craig M. Thompson, Itamar Simon, Julia Zeitlinger, Ezra G. Jennings, Heather L. Murray, D. Benjamin Gordon, Bing Ren, John J. Wyrick, Jean-Bosco Tagne, Thomas L. Volkert, Ernest Fraenkel, David K. Gifford, and Richard A. Young. Transcriptional regulatory networks in <i>saccharomyces cerevisiae</i>. *Science*, 298(5594):799–804, 2002. doi: 10.1126/science.1075090. URL https://www.science.org/doi/abs/10.1126/science.1075090.

[41] Claudia Skok Gibbs, Christopher A Jackson, Giuseppe-Antonio Saldi, Andreas Tjärnberg, Aashna Shah, Aaron Watters, Nicholas De Veaux, Konstantine Tchourine, Ren Yi, Tymor Hamamsy, Dayanne M Castro, Nicholas Carriero, David Gresham, Emily R Miraldi, and Richard Bonneau. High performance single-cell gene regulatory network inference at scale: The inferelator 3.0. *bioRxiv*, 2021. doi: 10.1101/2021.05.03.442499. URL https://www.biorxiv.org/content/early/2021/08/26/2021.05.03.442499.

[42] Miguel Cacho Teixeira, Romeu Viana, Margarida Palma, Jorge Oliveira, Mónica Galocha, Marta Neves Mota, Diogo Couceiro, Maria Galhardas Pereira, Miguel Antunes, Inês V Costa, Pedro Pais, Carolina Parada, Claudine Chaouiya, Isabel Sá-Correia, and Pedro Tiago Monteiro. YEASTRACT+: a portal for the exploitation of global transcription regulation and metabolic model data in yeast biotechnology and pathogenesis. *Nucleic Acids Res.*, November 2022.

[43] Richard Anderson-Sprecher. Model comparisons and r 2. *The American Statistician*, 48(2):113–117, 1994. doi: 10.1080/00031305.1994.10476036. URL https://doi.org/10.1080/00031305.1994.10476036.

[44] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL http://www.deeplearningbook.org.

[45] Jürg Bähler. Cell-cycle control of gene expression in budding and fission yeast. *Annu. Rev. Genet.*, 39: 69–94, 2005.

[46] Shadia Zaman, Soyeon Im Lippman, Xin Zhao, and James R Broach. How saccharomyces responds to nutrients. *Annu. Rev. Genet.*, 42:27–81, 2008.

[47] M Kanehisa and S Goto. KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, 28(1): 27–30, January 2000.

[48] Yuhan Hao, Stephanie Hao, Erica Andersen-Nissen, William M. Mauck, Shiwei Zheng, Andrew Butler, Maddie J. Lee, Aaron J. Wilk, Charlotte Darby, Michael Zager, Paul Hoffman, Marlon Stoeckius, Efthymia Papalexi, Eleni P. Mimitou, Jaison Jain, Avi Srivastava, Tim Stuart, Lamar M. Fleming, Bertrand Yeung, Angela J. Rogers, Juliana M. McElrath, Catherine A. Blish, Raphael Gottardo, Peter Smibert, and Rahul Satija. Integrated analysis of multimodal single-cell data. *Cell*, 184(13):3573–3587.e29, 2021. ISSN 0092-8674. doi: https://doi.org/10.1016/j.cell.2021.04.048. URL https://www.sciencedirect.com/science/article/pii/S0092867421005833.

[49] Vincent Traag, Ludo Waltman, and Nees Jan van Eck. From louvain to leiden: guaranteeing well-connected communities, 2018.

[50] Sabina Sangaletti, Claudio Tripodo, Claudia Chiodoni, Carla Guarnotta, Barbara Cappetti, Patrizia Casalini, Silvia Piconese, Mariella Parenza, Cristiana Guiducci, Caterina Vitali, and Mario P. Colombo. Neutrophil extracellular traps mediate transfer of cytoplasmic neutrophil antigens to myeloid dendritic cells toward ANCA induction and associated autoimmunity. *Blood*, 120(15):3007–3018, 10 2012. ISSN 0006-4971. doi: 10.1182/blood-2012-03-416156. URL https://doi.org/10.1182/blood-2012-03-416156.

[51] Qi Luo, Xiaojing Ma, Sharon M. Wahl, James J. Bieker, Merlin Crossley, and Luis J. Montaner. Activation and repression of interleukin-12 p40 transcription by erythroid kruppel-like factor in macrophages *. *Journal of Biological Chemistry*, 279(18):18451–18456, Apr 2004. ISSN 0021-9258. doi: 10.1074/jbc.M400320200. URL https://doi.org/10.1074/jbc.M400320200.

[52] César Cobaleda, Alexandra Schebesta, Alessio Delogu, and Meinrad Busslinger. Pax5: the guardian of b cell identity and function. *Nature Immunology*, 8(5):463–470, May 2007. ISSN 1529-2916. doi: 10.1038/ni1454. URL https://doi.org/10.1038/ni1454.

[53] Nidhi Malhotra and Joonsoo Kang. Smad regulatory networks construct a balanced immune system. *Immunology*, 139(1):1–10, 2013. doi: https://doi.org/10.1111/imm.12076. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/imm.12076.

[54] Prashant Trikha, Jena E. Moseman, Aarohi Thakkar, Amanda R. Campbell, Ezgi Elmas, Jennifer A. Foltz, Nitin Chakravarti, James R. Fitch, Elaine R. Mardis, and Dean A. Lee. Defining the AHR-regulated transcriptome in NK cells reveals gene expression programs relevant to development and function. *Blood Advances*, 5(22):4605–4618, 11 2021. ISSN 2473-9529. doi: 10.1182/bloodadvances.2021004533. URL https://doi.org/10.1182/bloodadvances.2021004533.

[55] Shin-ichi Kano, Kojiro Sato, Yasuyuki Morishita, Sabine Vollstedt, Sunhwa Kim, Keith Bishop, Kenya Honda, Masato Kubo, and Tadatsugu Taniguchi. The contribution of transcription factor irf1 to the interferon-γ–interleukin 12 signaling axis and th1 versus th-17 differentiation of cd4+ t cells. *Nature Immunology*, 9(1):34–41, Jan 2008. ISSN 1529-2916. doi: 10.1038/ni1538. URL https://doi.org/10.1038/ni1538.

[56] Svetlana Korinfskaya, Sreeja Parameswaran, Matthew T. Weirauch, and Artem Barski. Runx transcription factors in t cells—what is beyond thymic development? *Frontiers in Immunology*, 12, 2021. ISSN 1664-3224. doi: 10.3389/fimmu.2021.701924. URL https://www.frontiersin.org/articles/10.3389/fimmu.2021.701924.

[57] Xindong Liu, Yun Wang, Huiping Lu, Jing Li, Xiaowei Yan, Minglu Xiao, Jing Hao, Andrei Alekseev, Hiep Khong, Tenghui Chen, Rui Huang, Jin Wu, Qiwen Zhao, Qi Wu, Senlin Xu, Xiaohu Wang, Wei Jin, Shicang Yu, Yan Wang, Lai Wei, Aibo Wang, Bo Zhong, Ling Ni, Xiaolong Liu, Roza Nurieva, Lilin Ye, Qiang Tian, Xiu-Wu Bian, and Chen Dong. Genome-wide analysis identifies nr4a1 as a key

mediator of t cell dysfunction. *Nature*, 567(7749):525–529, Mar 2019. ISSN 1476-4687. doi: 10.1038/s41586-019-0979-8. URL https://doi.org/10.1038/s41586-019-0979-8.

[58] Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M. Mauck III, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive integration of single-cell data. *Cell*, 177(7):1888–1902.e21, Jun 2019. ISSN 0092-8674. doi: 10.1016/j.cell.2019.05.031. URL https://doi.org/10.1016/j.cell.2019.05.031.

[59] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[60] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[61] Donald E. Hilt, Donald W. Seegrist, United States. Forest Service., and Northeastern Forest Experiment Station (Radnor. Pa.). *Ridge, a computer program for calculating ridge regression estimates*, volume no.236. Upper Darby, Pa, Dept. of Agriculture, Forest Service, Northeastern Forest Experiment Station, 1977, 1977. URL https://www.biodiversitylibrary.org/item/137258. https://www.biodiversitylibrary.org/bibliography/68934.

[62] J.V. Beck and K.J. Arnold. *Parameter Estimation in Engineering and Science*. Probability and Statistics Series. Wiley, 1977. ISBN 9780471061182. URL https://books.google.com/books?id=_qAYgYN87UQC.

[63] GH Golub, Michael Heath, and Grace Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 1979. URL http://amstat.tandfonline.com/doi/abs/10.1080/00401706.1979.10489751.

[64] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

[65] Christine Vogel and Edward M. Marcotte. Insights into the regulation of protein abundance from proteomic and transcriptomic analyses. *Nature Reviews Genetics*, 13:227 EP –, Mar 2012. URL https://doi.org/10.1038/nrg3185.

[66] Pierre Nicolas, Ulrike Mäder, Etienne Dervyn, Tatiana Rochat, Aurélie Leduc, Nathalie Pigeonneau, Elena Bidnenko, Elodie Marchadier, Mark Hoebeke, Stéphane Aymerich, et al. Condition-dependent transcriptome reveals high-level regulatory architecture in bacillus subtilis. *Science*, 335(6072):1103–6, 2012.

[67] Konstantine Tchourine, Christine Vogel, and Richard Bonneau. Condition-specific modeling of biophysical parameters advances inference of regulatory networks. *Cell Reports*, 23(2):376 – 388, 2018. ISSN 2211-1247. doi: https://doi.org/10.1016/j.celrep.2018.03.048. URL http://www.sciencedirect.com/science/article/pii/S2211124718303930.

[68] Sean R Hackett, Edward A Baltz, Marc Coram, Bernd J Wranik, Griffin Kim, Adam Baker, Minjie Fan, David G Hendrickson, Marc Berndl, and R Scott McIsaac. Learning causal networks using inducible transcription factors and transcriptome-wide time series. *Molecular Systems Biology*, 16(3):e9174, 2020. doi: 10.15252/msb.20199174. URL https://www.embopress.org/doi/abs/10.15252/msb.20199174.

[69] Pedro T Monteiro, Jorge Oliveira, Pedro Pais, Miguel Antunes, Margarida Palma, Mafalda Cavalheiro, Mónica Galocha, Cláudia P Godinho, Luís C Martins, Nuno Bourbon, Marta N Mota, Ricardo A Ribeiro, Romeu Viana, Isabel Sá-Correia, and Miguel C Teixeira. YEASTRACT+: a portal for cross-species comparative genomics of transcription regulation in yeasts. *Nucleic Acids Res.*, 48(D1):D642–D649, January 2020.

[70] Abbas Jariani, Lieselotte Vermeersch, Bram Cerulus, Gemma Perez-Samper, Karin Voordeckers, Thomas Van Brussel, Bernard Thienpont, Diether Lambrechts, and Kevin J Verstrepen. A new protocol for single-cell RNA-seq reveals stochastic gene expression during lag phase in budding yeast. *Elife*, 9, May 2020.

[71] P T Spellman, G Sherlock, M Q Zhang, V R Iyer, K Anders, M B Eisen, P O Brown, D Botstein, and B Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell*, 9(12):3273–3297, December 1998.

[72] Audrey P Gasch, Feiqiao Brian Yu, James Hose, Leah E Escalante, Mike Place, Rhonda Bacher, Jad Kanbar, Doina Ciobanu, Laura Sandor, Igor V Grigoriev, Christina Kendziorski, Stephen R Quake, and Megan N McClean. Single-cell RNA sequencing reveals intrinsic and extrinsic regulatory heterogeneity in yeast responding to stress. *PLoS Biol.*, 15(12):e2004050, December 2017.

[73] Anna Danese, Maria L. Richter, Kridsadakorn Chaichoompu, David S. Fischer, Fabian J. Theis, and Maria Colomé-Tatché. Episcanpy: integrated single-cell epigenomic analysis. *Nature Communications*, 12(1):5228, Sep 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-25131-3. URL https://doi.org/10.1038/s41467-021-25131-3.

[74] F. Alexander Wolf, Philipp Angerer, and Fabian J. Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1):15, Feb 2018. ISSN 1474-760X. doi: 10.1186/s13059-017-1382-0. URL https://doi.org/10.1186/s13059-017-1382-0.

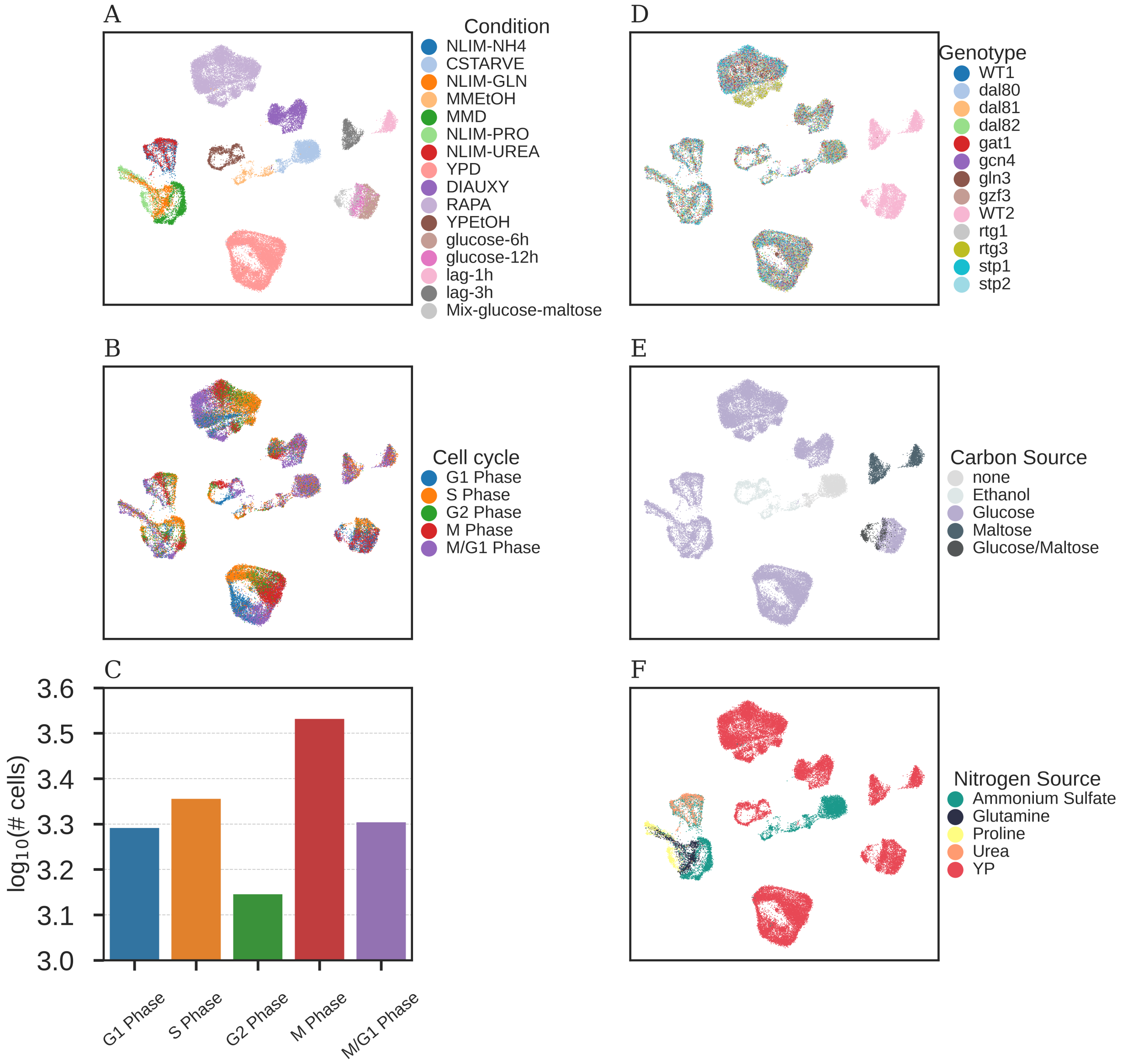
[75] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

[76] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3): 90–95, 2007. doi: 10.1109/MCSE.2007.55.

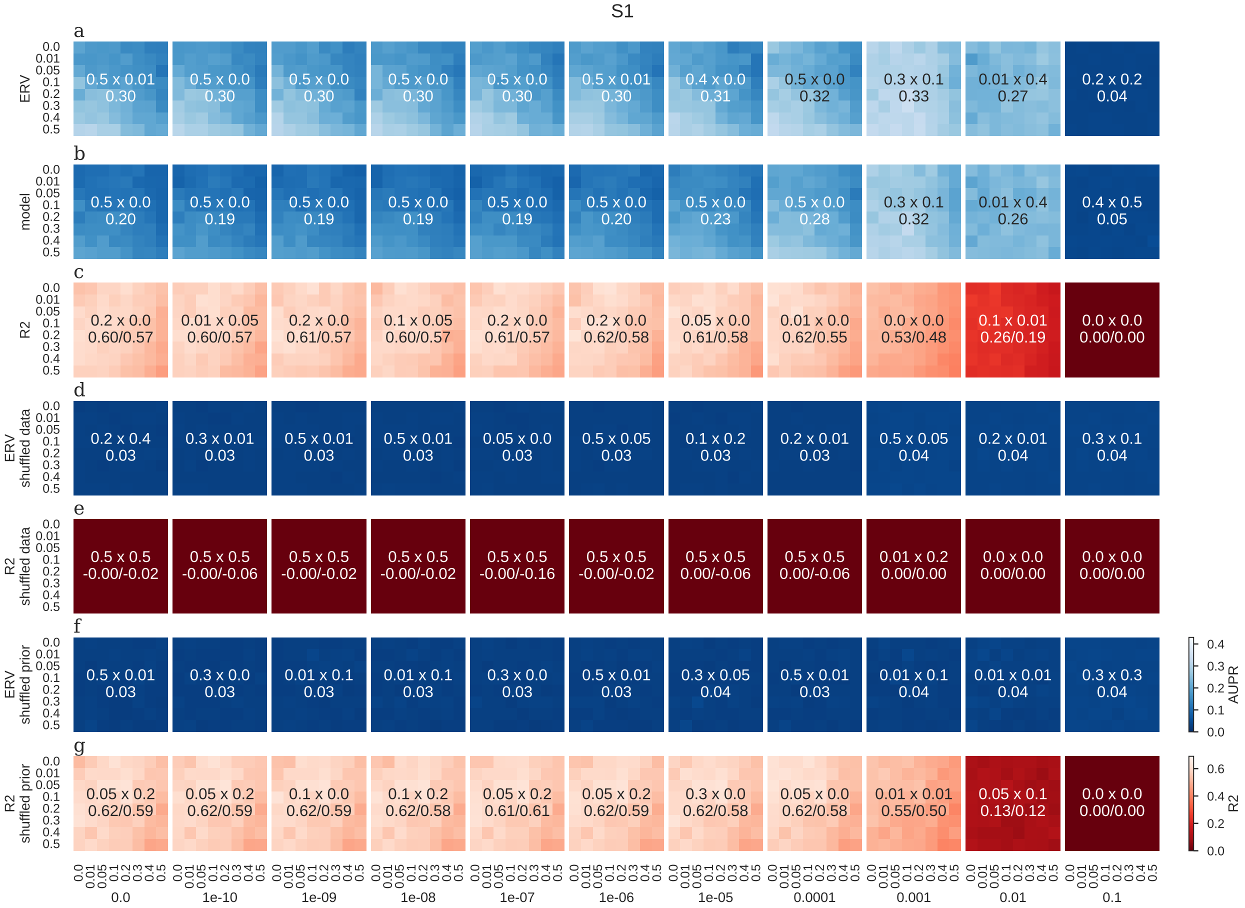
[77] Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60): 3021, 2021. doi: 10.21105/joss.03021. URL https://doi.org/10.21105/joss.03021.

[78] Abbas Jariani, Lieselotte Vermeersch, Bram Cerulus, Gemma Perez-Samper, Karin Voordeckers, Thomas Van Brussel, Bernard Thienpont, Diether Lambrechts, and Kevin J Verstrepen. A new protocol for single-cell rna-seq reveals stochastic gene expression during lag phase in budding yeast. *eLife*, 9:e55320, may 2020. ISSN 2050-084X. doi: 10.7554/eLife.55320. URL https://doi.org/10.7554/eLife.55320.

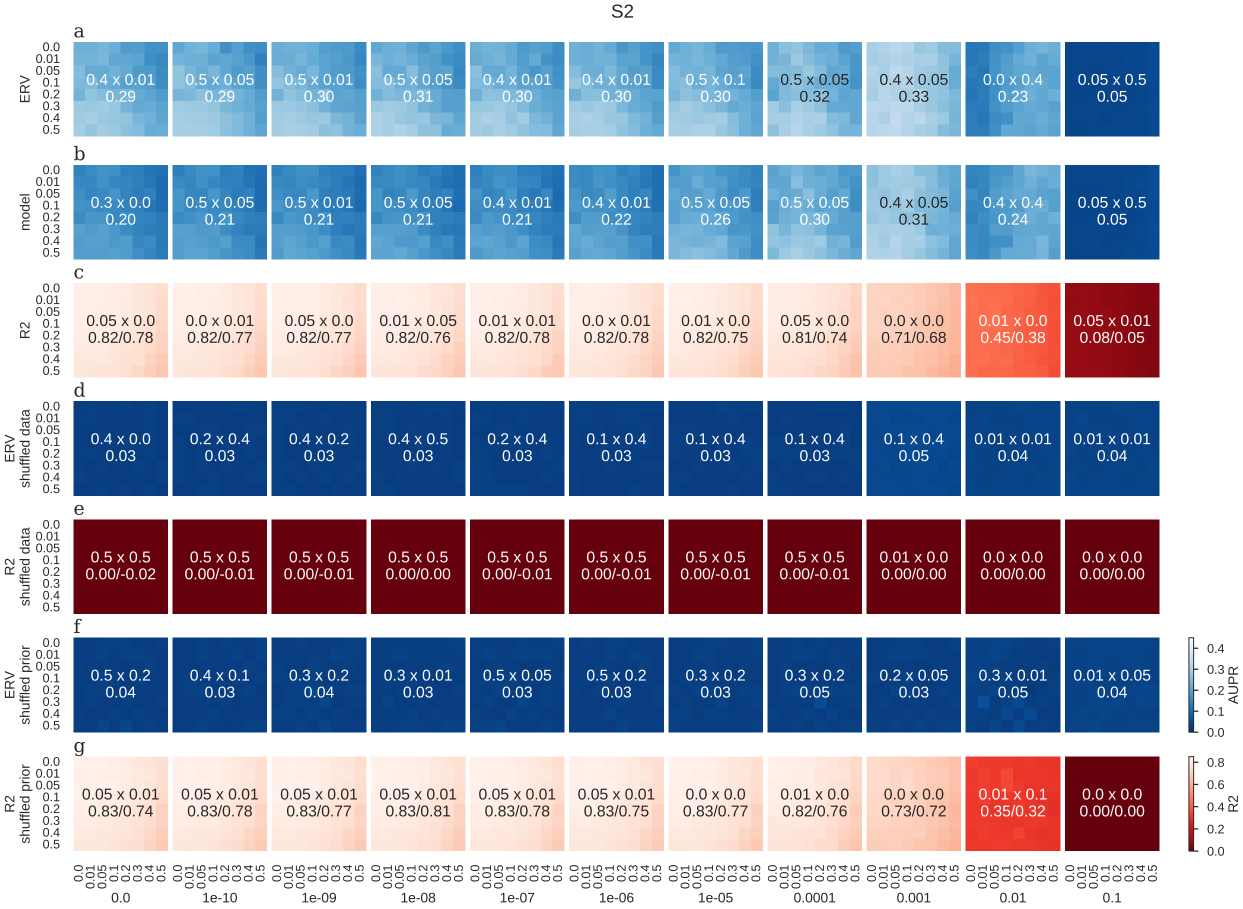
[79] Leland McInnes and John Healy. Umap: Uniform manifold approximation and projection for dimension reduction. *ArXiv e-prints*, feb 2018.



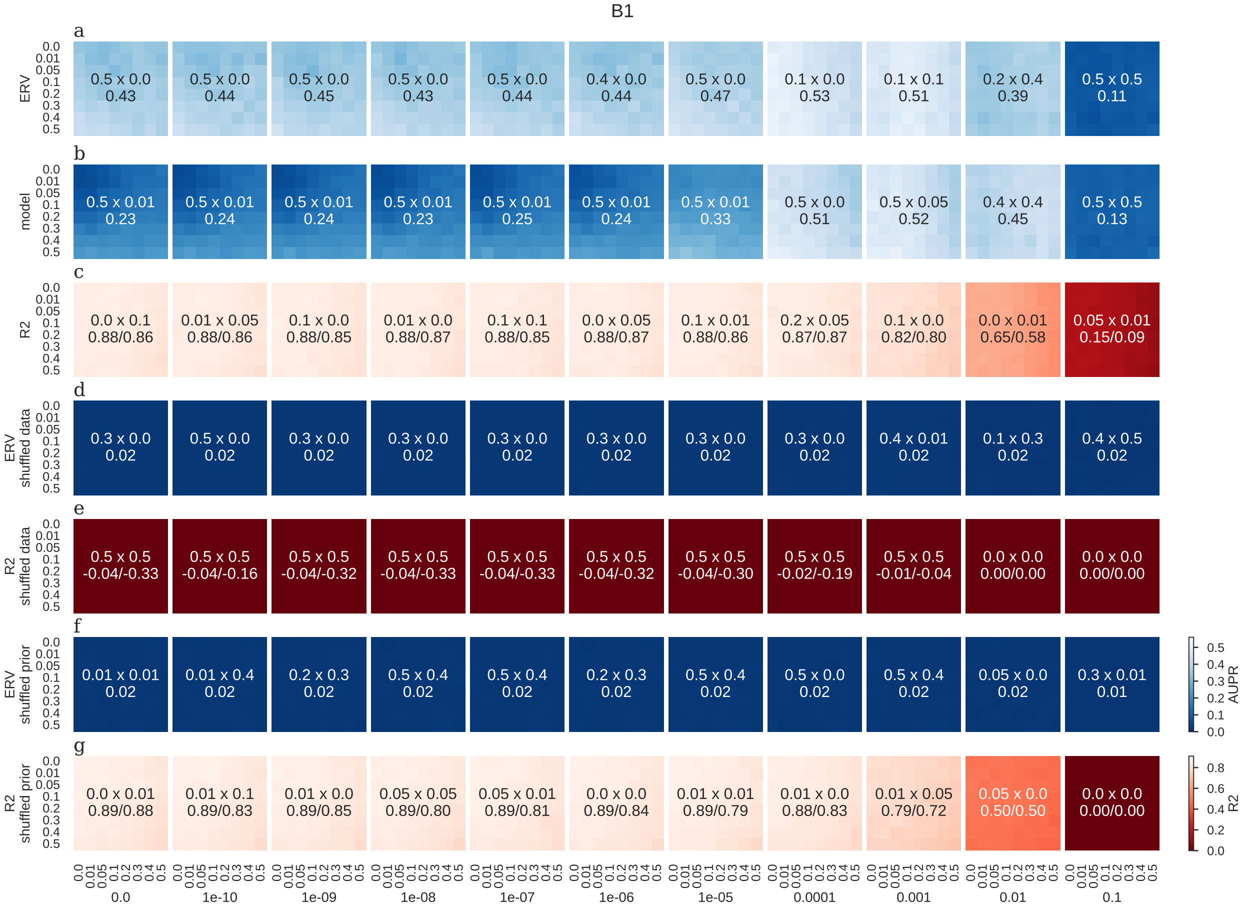
S1: Overview of the single cell combined datasets [13] and [78] (scY) in *S. cerevisiae*. UMAP [79] projected data with annotation of **A**: growth condition, **B**: inferred cell cycle phase, and **C**: Frequency of cell cycle annotated cells in YPD. **D**: Genotypes. **E**: Carbon source for growth condition. **F**: Nitrogen source for growth condition.



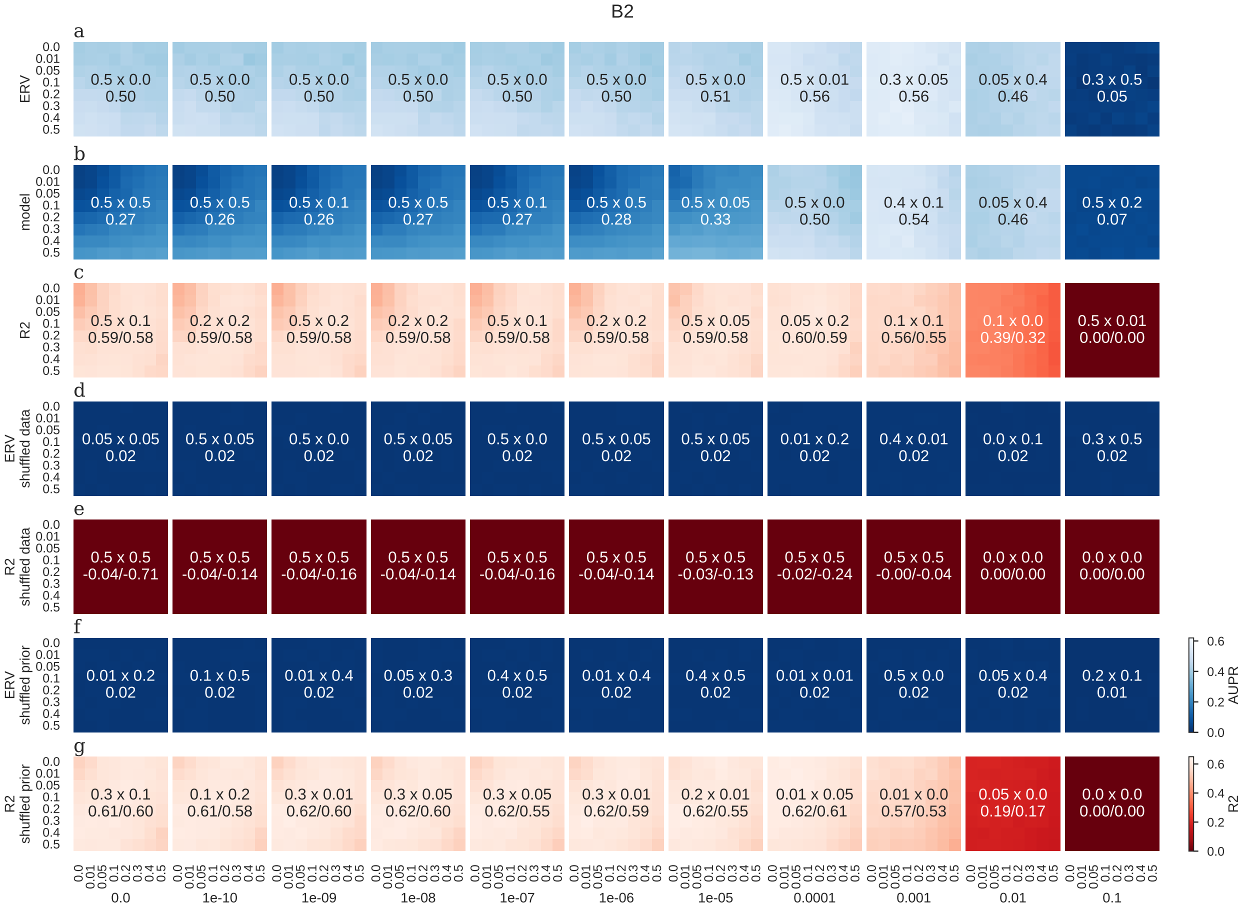
S2: Dataset S1[68]. Hyper-parameter cube search. Each column of squares indicate a weight-decay (L2) penalty, with 0 and then 10 log linear values between $10^{-10} - 10^{-1}$. Each square is the grid search of dropout on input feature (y-axis) and latent features (x-axis). Rows in blue shade (**a**, **b**, **d** and **f**) indicate GRN recovery performance measured by area under precision recall curve (AUPR) on 20% held out of gold-standard network. ERV (**a**, **d** and **f**) indicate the GRN is ranking based on explained relative variance (Section 4.4.1), model (**b**) indicate network interactions are ranked based on their weights magnitude, with larger magnitude indicate more influential. Red shades (**c**, **e** and **g**) indicate prediction accuracy R^2 on held out, n=50%, of samples. Negative controls are done by shuffling either the data (**d** and **e**) or the prior network structure (**f** and **g**).



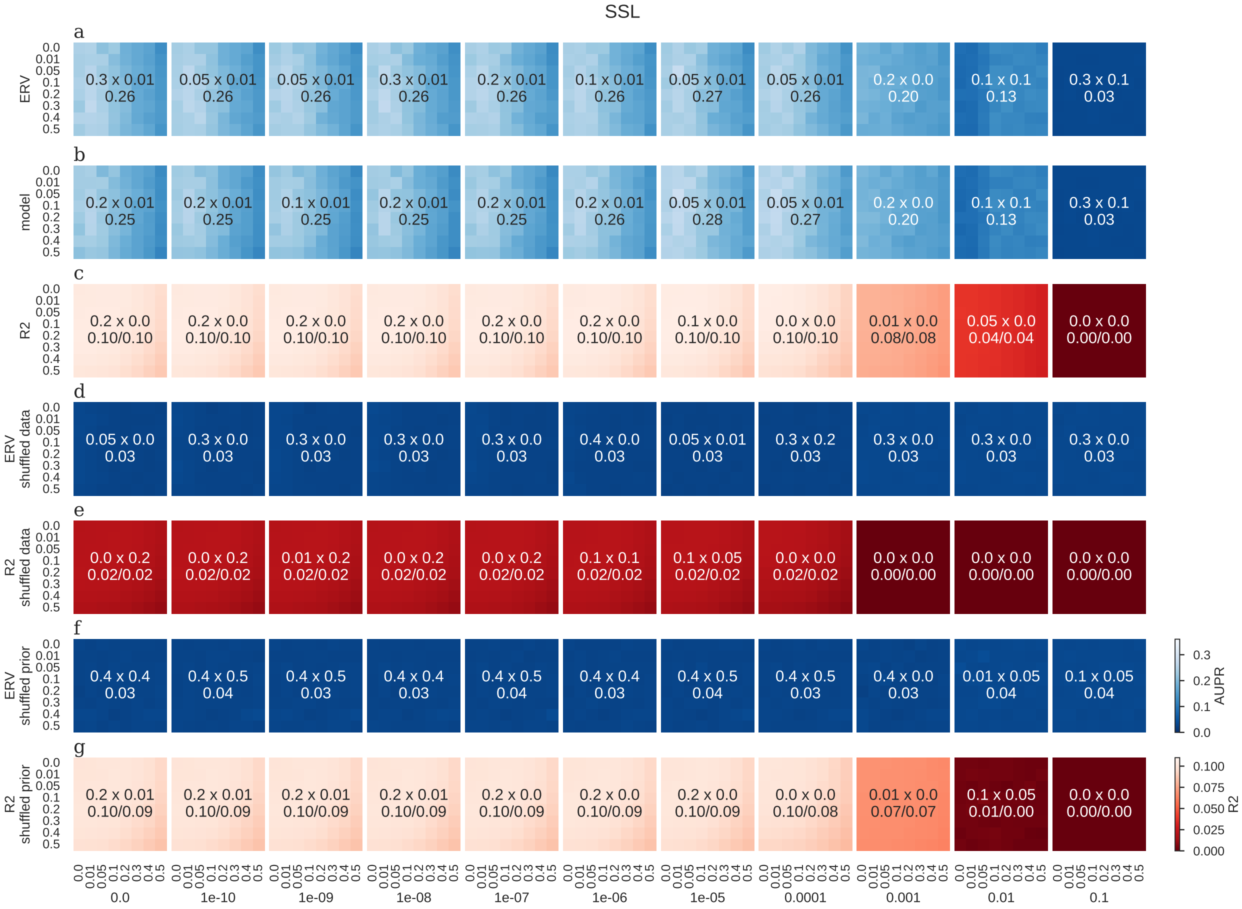
S3: Dataset S2[67]. Description as in Figure S2



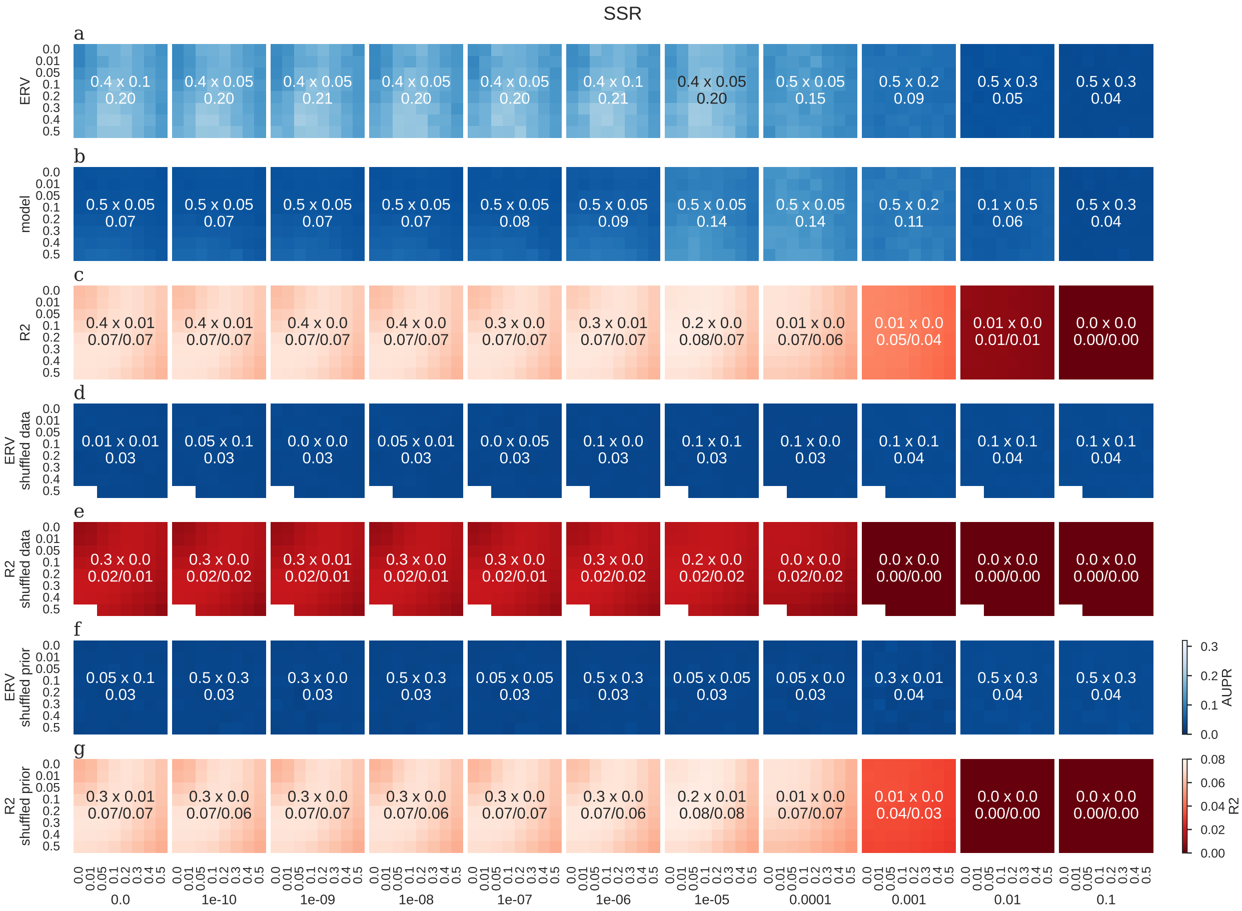
S4: Dataset B1[10]. Description as in Figure S2



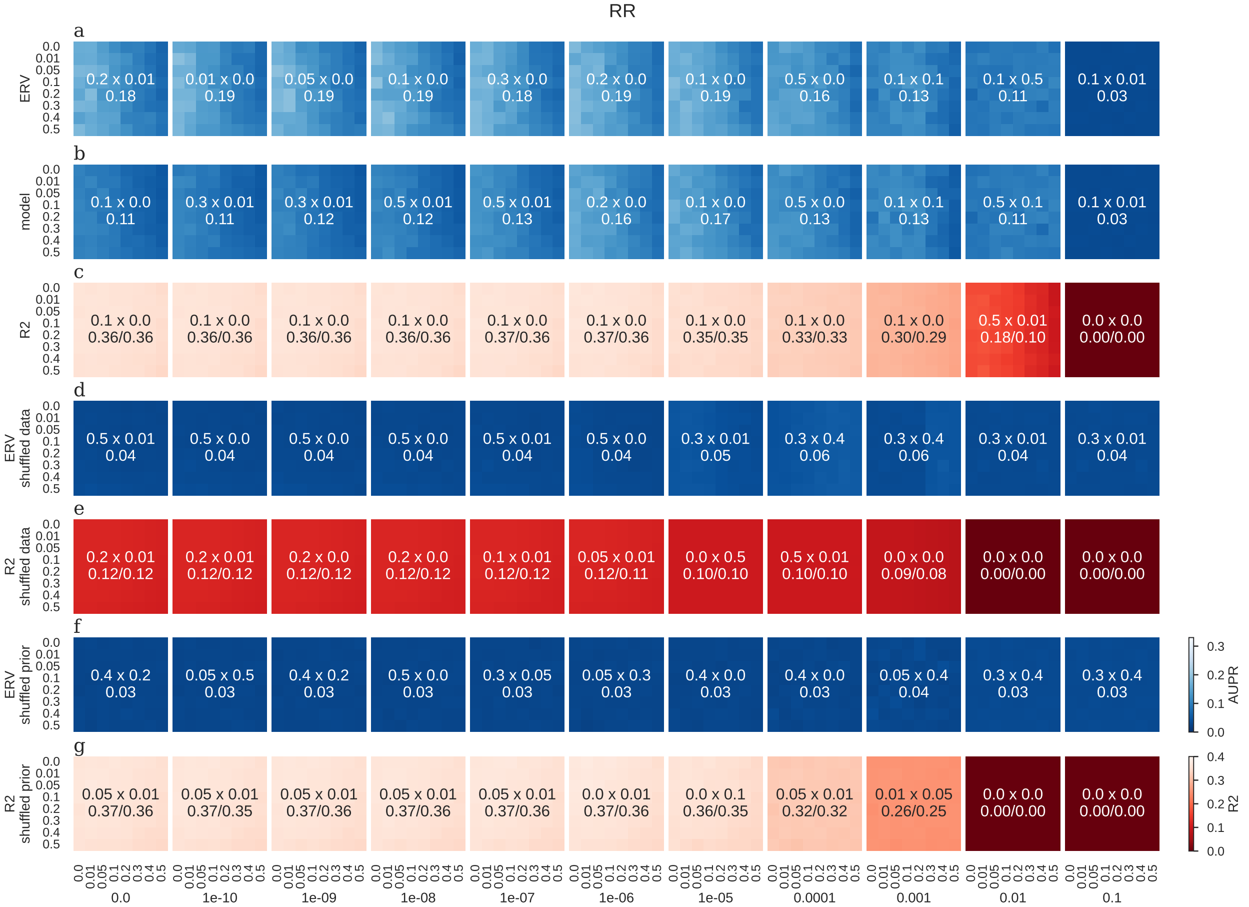
S5: Dataset B2[66]. Description as in Figure S2



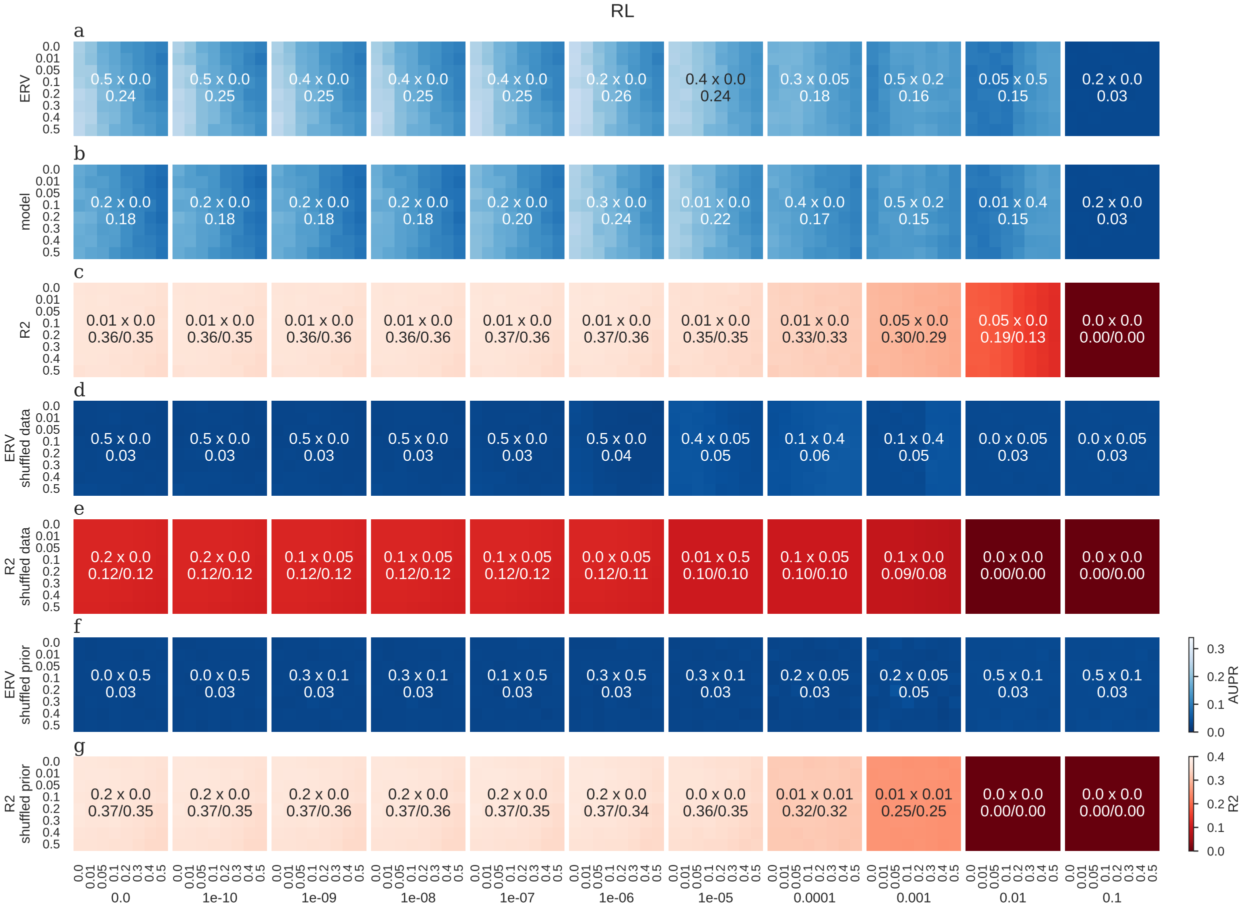
S6: Dataset scY[13, 78] with StandardScaler normalisation and linear activation (SSL) configuration. Description as in Figure S2



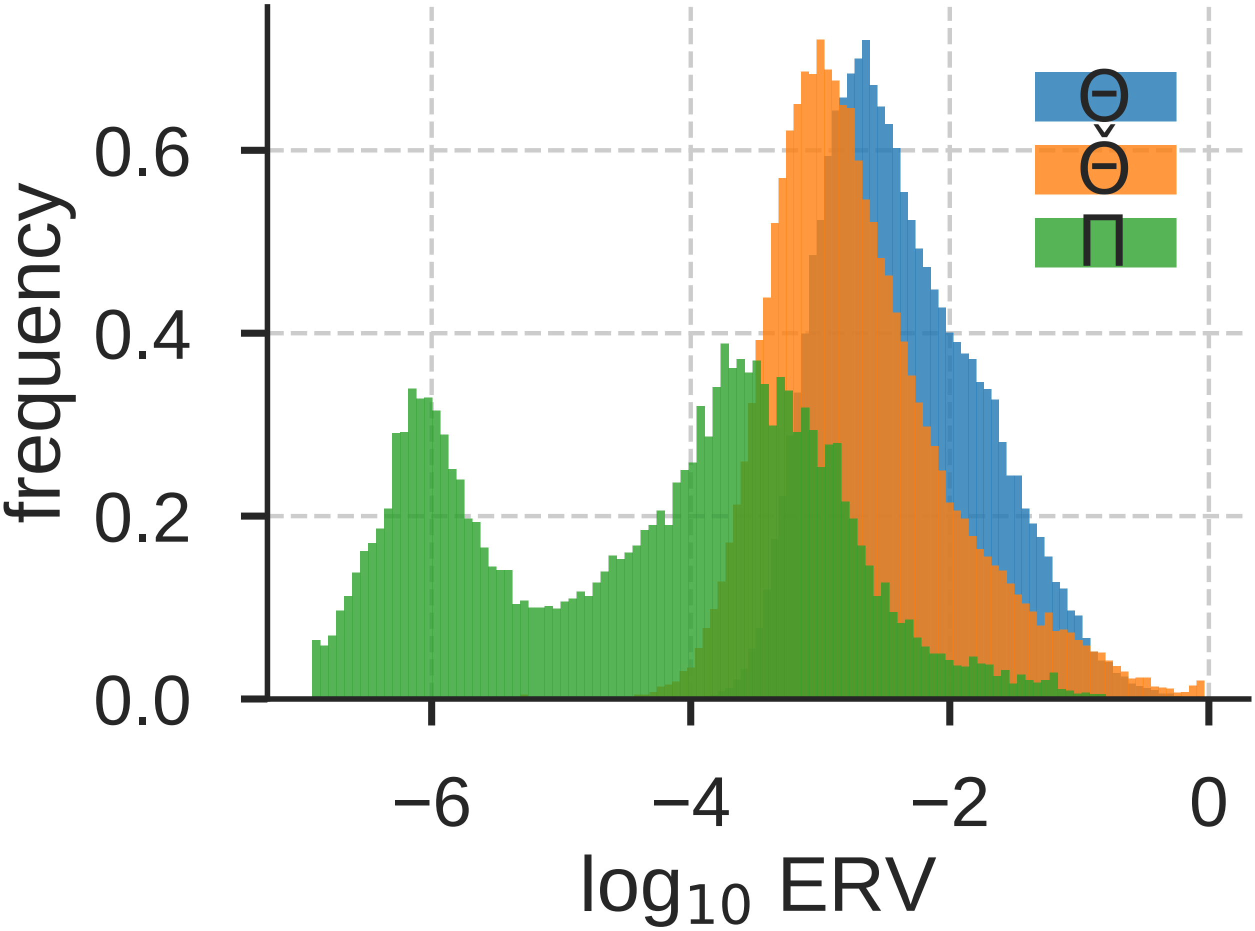
S7: Dataset scY[13, 78] with StandardScaler normalisation ReLU$_0$ activation (SSR). Description as in Figure S2



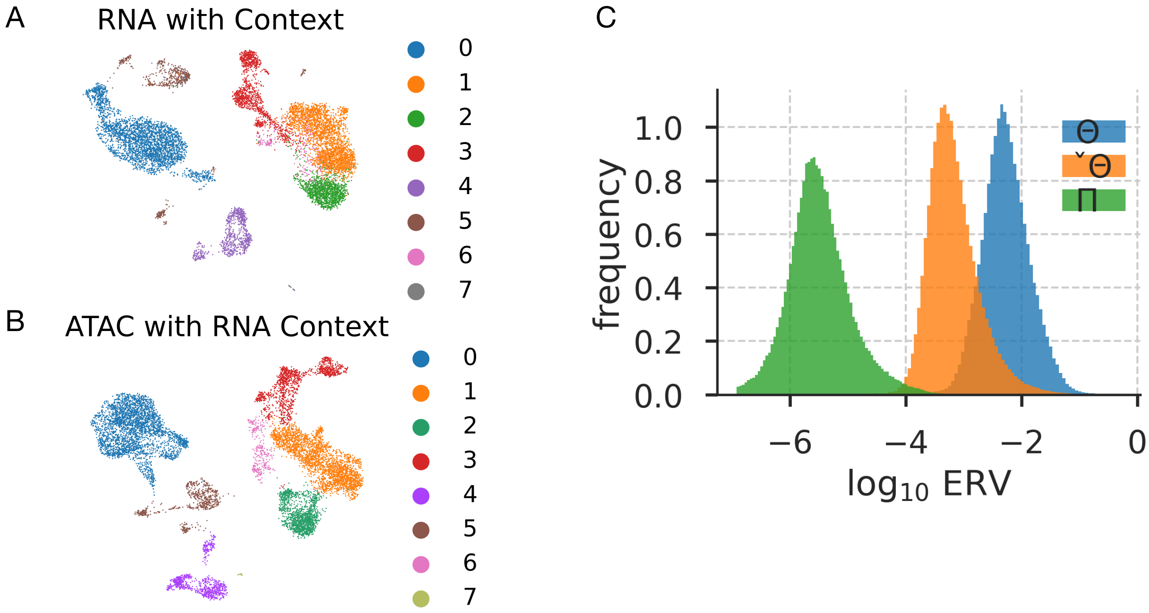
S8: Dataset scY[13, 78] with RobustMinScaler normalisation and ReLU_0 activation (RR). Description as in Figure S2



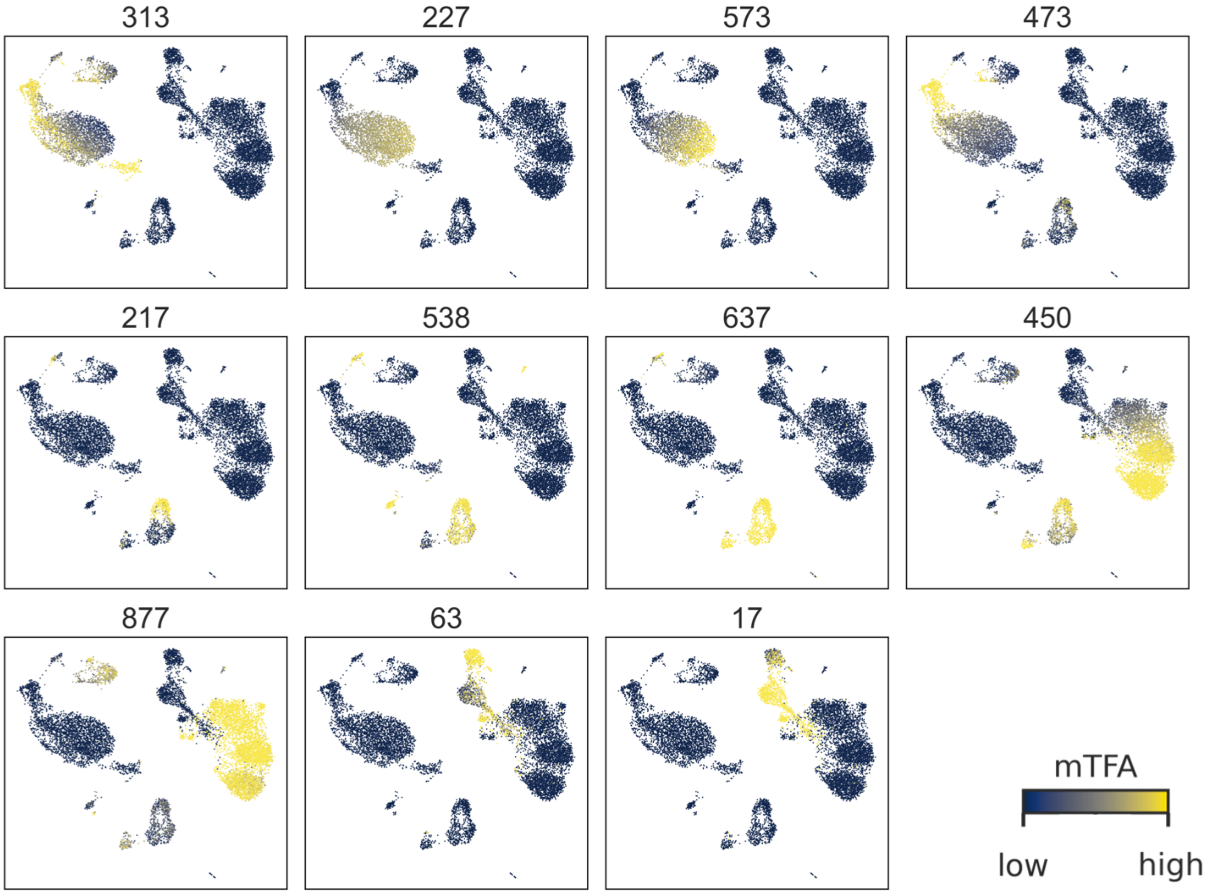
S9: Dataset scY[13, 78] with RobustMinScaler normalisation and linear activation (RL). Description as in Figure S2



S10: Distribution of ERV (ξ^2) for hierarchical SupirFactor trained on scY.



S11: Single cell PBMC Contexts and Networks **A**: UMAP projection of the scRNA-Seq PBMC dataset with "Context" via leiden clustering (0.2) **B**: UMAP projection of the scATAC-Seq PBMC dataset with "Context" via leiden clustering (0.2) **C**: Distribution of ERV for $\hat{\mathbf{\Theta}}$, $\mathbf{\Pi}$ and $\mathbf{\Theta}$ networks



S12: UMAP projection of all functionally enriched mTF activation (mTF activity scaled [0, 1] for comparison).