

RESEARCH

Open Access



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

Andreas Tjärnberg^{1,2,3,8,11*} , Maggie Beheler-Amass^{2,3†}, Christopher A. Jackson^{2,3†}, Lionel A. Christiaen^{1,3,9,10}, David Gresham^{2,3} and Richard Bonneau^{2,3,4,5,6,7*}

[†]Maggie Beheler-Amass and Christopher A. Jackson contributed equally to this work.

*Correspondence: andreas.tjarnberg@fripost.org; bonneau.richard@gene.com

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
Full list of author information is available at the end of the article

Abstract

Background: Modeling of gene regulatory networks (GRNs) is limited due to a lack of direct measurements of genome-wide transcription factor activity (TFA) making it difficult to separate covariance and regulatory interactions. Inference of regulatory interactions and TFA requires aggregation of complementary evidence. Estimating TFA explicitly is problematic as it disconnects GRN inference and TFA estimation and is unable to account for, for example, contextual transcription factor-transcription factor interactions, and other higher order features. Deep-learning offers a potential solution, as it can model complex interactions and higher-order latent features, although does not provide interpretable models and latent features.

Results: We propose a novel autoencoder-based framework, *StrUcture Primed Inference of Regulation using latent Factor ACTivity* (SupirFactor) for modeling, and a metric, explained relative variance (ERV), for interpretation of GRNs. We evaluate SupirFactor with ERV in a wide set of contexts. Compared to current state-of-the-art GRN inference methods, SupirFactor performs favorably. We evaluate latent feature activity as an estimate of TFA and biological function in *S. cerevisiae* as well as in peripheral blood mononuclear cells (PBMC).

Conclusion: Here we present a framework for structure-primed inference and interpretation of GRNs, SupirFactor, demonstrating interpretability using ERV in multiple biological and experimental settings. SupirFactor enables TFA estimation and pathway analysis using latent factor activity, demonstrated here on two large-scale single-cell datasets, modeling *S. cerevisiae* and PBMC. We find that the SupirFactor model facilitates biological analysis acquiring novel functional and regulatory insight.



© The Author(s) 2024. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>. The Creative Commons Public Domain Dedication waiver (<http://creativecommons.org/publicdomain/zero/1.0/>) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

Background

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 and 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 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) can be thought of as a representation of how much a transcription factor influences the transcriptional rate of Pol II at its target genes, theoretically measured as the marginal rate of nascent transcripts produced per minute per target gene. To have activity, the TF protein must itself be expressed, and it must be able to localize to the nucleus and bind to its DNA target. Depending on the TF, it may require specific post-translational modifications, presence or absence of a small molecule, and presence or absence of co-activators or co-repressors. This TF activity is difficult to measure experimentally genome-wide.

Explicit inference of 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 remain an issue.

Using more complex models to better match known transcriptional regulatory biology places numerous demands on optimization and inference machineries and limits scale. However, the field of deep learning offers scalable learning and optimization techniques that could aid in GRN inference. Deep learning has 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 and is an active area of research [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 methods often require a

target value, e.g., as in classification tasks, or cannot be compared reliably between output features as non-normalized error rates depend on feature scaling. These techniques cannot be used for model selection. To make statements on the active GRN in a given context, we need to be able to determine predictive and non-predictive interactions. For unbounded metrics, this remains a problem as we need to construct a secondary heuristic on what constitute predictive and non-predictive bounds. Deep learning models generally remain over-parameterized, making biological interpretation difficult, and techniques must be applied after training to eliminate model parameters and enforce sparsity for GRN evaluation [27–35].

Knowledge priming embeds existing structural evidence into the model architecture by limiting connectivity between features based on epigenetic or regulatory evidence found in the literature. This informs the model of constraints on regulation which are complementary and not directly measured by the readout of gene expression in the training data. These constraints allow direct interpretation of the model, as priming with biological interactions allows biological interpretation of the resulting network graph [36]. Other structure evidence embedding methods use this prior data type as a constraint on covariance [37]. Both of these methods lack the ability to infer novel regulatory structure.

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*, genes that have known upstream regulators given an independent set of evidence derived from literature or epigenetic data as described in “[The SupirFactor regulation model](#)” section, and allows the activation of latent features in 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, which 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.

The SupirFactor regulation model

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

We actualize this model using a deep learning framework choosing f , and g and learn \mathbf{W} , and Θ with ϕ as a latent feature produced by the weighted output from \mathbf{x} mapped through a learned \mathbf{W} . Depending on the form of f , and including non-linearities, we can learn additional higher order interactions and regulatory pathways (see the “[StrUcture Primed Inference of Regulation using latent Factor ACTivity \(SupirFactor\)](#)” section). The complex version of the model that can capture other interactions in f we call “hierarchical” SupirFactor (Fig. 1A). A simple version of this framework uses a single bottleneck layer for our function f we refer to as “shallow” SupirFactor (Fig. 1B).

Constructing a prior matrix \mathbf{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 \mathbf{P} is likely to include more false positives and will therefore result in a noisier propagation of TF variance. A sparser \mathbf{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 \mathbf{P} rarely includes reliable sign or weight estimates. Inferring signs for \mathbf{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 $\mathbf{W} \in \mathbf{P}$ dependant on Θ will mitigate these problems.

Selection of SupirFactor hyperparameters

We evaluate the SupirFactor framework (Fig. 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 *Bacillus subtilis* gold standard network [10], covering 154

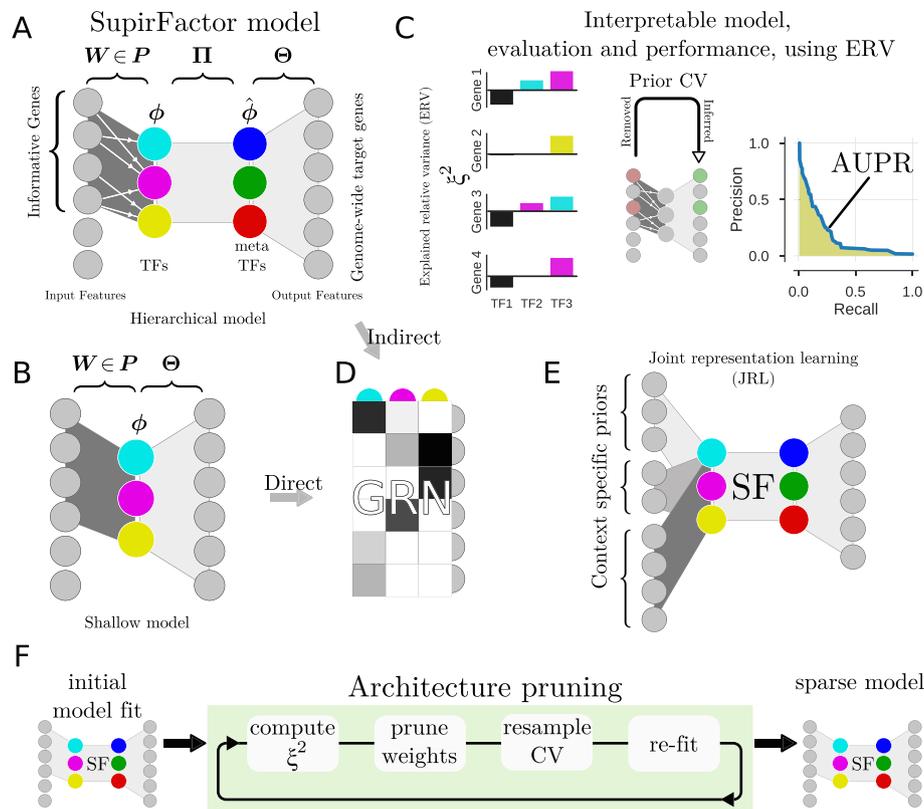


Fig. 1 Outline of the SupirFactor framework. The SupirFactor model is constructed like an autoencoder where we embed gene expression data on the transcription factor manifold, exploring two architectures, the “hierarchical” **A**, and the shallow **B** architecture. The output of the first layer defines the latent features marked as TFs (Transcription Factors) and the activation ϕ is the transcription factor activity (TFA). The prior P connect the evidence of TF to a set of informative downstream genes, with learnable weights W . For **A**, Π connects the TFs to the latent features, here called the meta TFs (mTFs). Θ weights the mTF activity (mTFA) to predict genome wide gene expression profiles. In **B**, the TFs directly weights TF to gene influence in Θ . **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 W and predicted in the GRN which for the shallow model is Θ and for the hierarchical model is $\hat{\Theta}$ 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 stabilize and eliminate over-parameterization by eliminating non-predicting model parameters facilitated by ERV

TFs and 4218 informative genes also used as the prior evidence, and a *S. cerevisiae* gold standard network [42], covering 98 TFs and 993 target genes, both derived from literature databases, we also use a *S. cerevisiae* prior evidence matrix with 150 TFs and 5578 informative genes used in previous work [41]. Single-cell RNA expression data is preprocessed (see the “Single cell pre-processing” section) and both bulk and single-cell data is feature normalized (see the “Feature normalization” section) prior to model fitting. For the SupirFactor model, the number of latent features used $|\Phi| =$ number of TFs available in the prior evidence.

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 (Fig. 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 disconnect each latent model feature and compute the consequent effect on the output variance MSE, scoring the separated model feature against the full model prediction without retraining (see the “[Model selection and model parameter ranking](#)” section).

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 generalization (see the “[Model regularization](#)” section). 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 (Additional file 1: Figs. 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 (Additional file 1: Figs. S3 & S4). We interpret this as overfitting to the prior network structure and increasing recovery of the gold standard in cases where these two structures align, such as in scale-free networks with dense highly connected TFs. The model then emphasizes these TFs at the cost of prediction accuracy and inclusion of less connected TFs. This is undesirable, so we select hyperparameters that maximizes 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 maximized in the span $L2 \in [10^{-6}, 10^{-4}]$ 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 (Additional file 1: Figs. 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 .

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 (Fig. 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

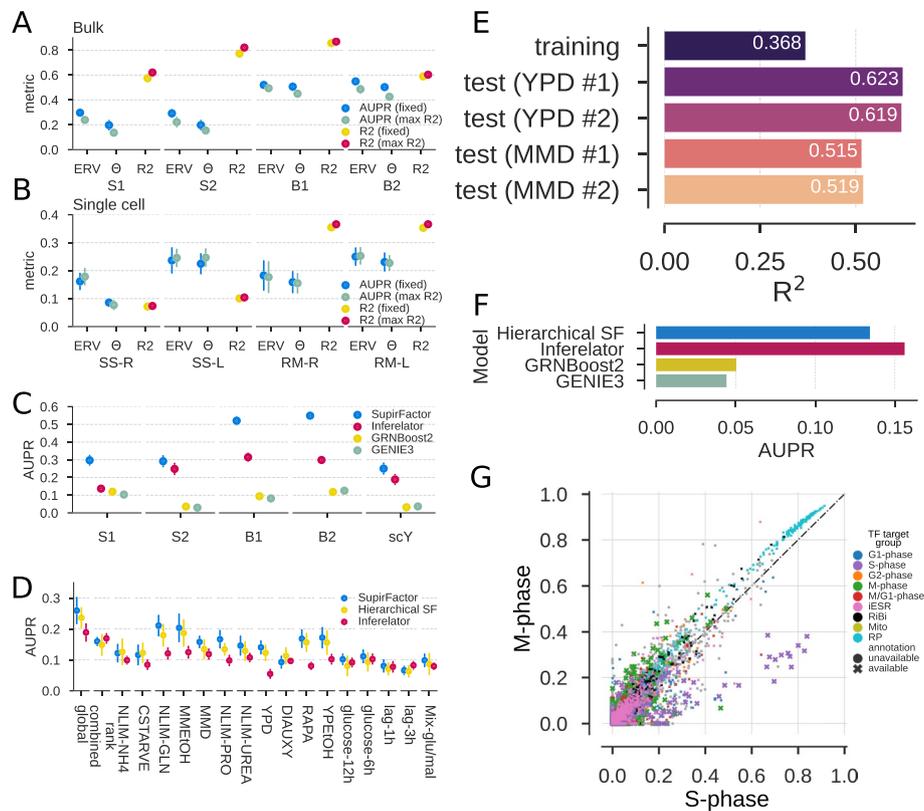


Fig. 2 SupirFactor benchmark and hyperparameter evaluation. Performance is evaluated on gold-standard networks consisting only of edges held out of the prior network \mathcal{P} , measuring recovery using AUPR. R^2 is computed on a validation set of 50% of the data samples held out of the training data. Data sets are labeled for species, with *Bacillus 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 \mathcal{P} . R^2 is calculated from the full network model. Model hyperparameters are set based on results in Additional file 1: Figs. S2-S9 as detailed in the “Model regularization” section. **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, and two methods not using prior evidence; GRNBoost2 and GENIE3. **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** Recovery of independently collected regulatory evidence not in the prior. Comparing the full Hierarchical SF with the Inferelator, GRNBoost2, and GENIE3 on using the trained single cell Yeast (scY) models. **G** 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 contextual 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

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 (Fig. 2B), we extend the comparisons, comparing linear and rectified linear (ReLU) activation functions (see the “ReLU in hierarchical SupirFactor” section), and comparing standard normalization to a robust sparsity-retaining normalization (see the “Feature normalization” section). RobustMinScaler outperforms StandardScaler normalization, implying that preserving sparse data for model training is advantageous (Fig. 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 (Fig. 2C), a method which takes comparable RNA expression and prior network inputs and learns a GRN. We note that other methods which take these inputs are not amenable to scoring on holdouts that we are employing for our benchmark [41], making comparison difficult. Adding prior evidence implies a causal chain defined by the prior, resulting in a better representation of regulation in the model. To test this assumption, we also benchmark against two state of the art methods using covariance of expression of a regulator to its targets, GRNBoost2 [37] and GENIE3 [34]. Using the configurations from our benchmarking, SupirFactor improves significantly over previous methods results on all of the data sets tested. Including prior evidence significantly improves network reconstruction.

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. We add an extra latent layer with number of latent features $|\hat{\Phi}| = |\Phi|$ as shown in Fig. 1A. 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 or change its regulatory role, e.g., activating or inhibiting, depending on its regulatory partners.

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 (Fig. S1A). We compare hierarchical SupirFactor (Fig. 1A), shallow SupirFactor (Fig. 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 (Fig. 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 (Fig. 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 neighboring 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.

Hierarchical SupirFactor introduces a larger set of learnable weights and a potential over-parameterization 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 attenuate over-parameterization (Fig. 1E). After 1 iteration of pruning, nonzero weights reduce to $\sim 57\% \in \Theta$, $\sim 52\% \in \hat{\Theta}$ and $\sim 80\% \in \Pi$ 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 \mathbf{P} without holdouts (Fig. 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 (Fig. 2E).

Finally, we wanted to validate on regulatory edges not in the prior. We assembled an independent set of regulatory edges (see the “Yeast test GRN” section) and removed the overlap with the prior used for training. With this independent set of regulatory evidence, we evaluated each method’s model performance with AUPR. We found that the prior based method models outperform the other methods tested for the independent set (Fig. 2F).

We conclude that SupirFactor generalizes 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 (Fig. 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 (Fig. 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 multi-subunit complexes, or by competing for the same DNA binding regions. These interactions

Table 1 Pairwise comparisons between context-specific cell cycle phase networks from hierarchical SuprFactor. 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

A	Context:		A		B		A		B		OR	p-val
	B	S	$\xi^2(A) > \xi^2(B)$	$\xi^2(B) > \xi^2(A)$	$\xi^2(A) > \xi^2(B)$	$\xi^2(B) > \xi^2(A)$	Total	Total	Total	Total		
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			

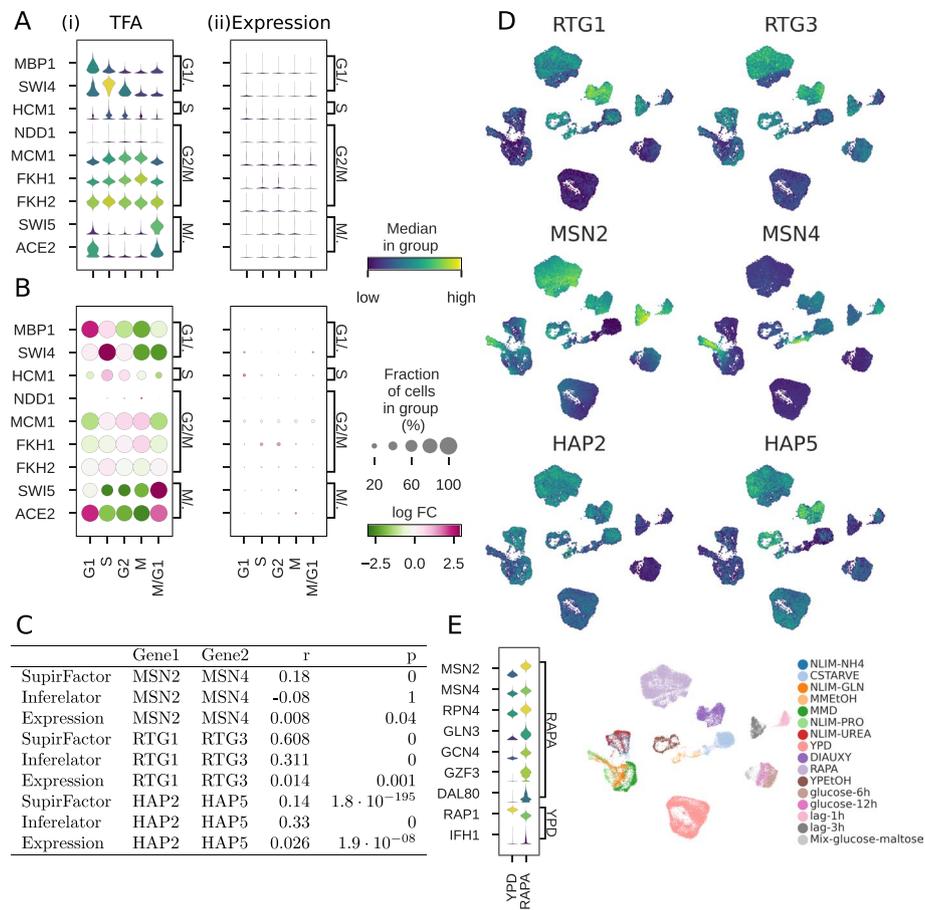


Fig. 3 Transcription factor activity in single cell yeast. TFA estimated from hierarchical SupirFactor model. Violin plots are generated by scaling [0, 1] the underlying measurement. **A** Cell cycle TFs regulate gene expression in specific cell cycle phases, with the phase the TF regulates annotated 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

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 (Fig. 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 (Fig. 3D).

Finally, we compare the TFA between perturbed, rapamycin-treated cells and untreated control cells (Fig. 3E). Rapamycin is expected to inhibit TOR pathway signaling, 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.

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. Forty-two of 148 mTFs are enriched for target genes (defined as mTF to gene connections with $\xi^2 > 0.1$) to these metabolic KEGG pathways (Fig. 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 data set use glucose as a primary carbon source (Additional file 1: Fig. S1E), and we see that these conditions have similar mTF activation (for mTFs; 78, 72, 57, 88, 64, 116) within the glucose conditions (Fig. 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 (Fig. 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 and 79) and which have heterogeneity within growth conditions (e.g., 57 and 78) (Fig. 4D). We observe that mTFs aggregate biologically functional groups in their targets and can be evaluated quantitatively as activities of these pathways.

SupirFactor model 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] (Fig. 5A). These two data types are integrated by using ATAC-seq chromatin accessibility as a cell-specific mask (Eq. 16).

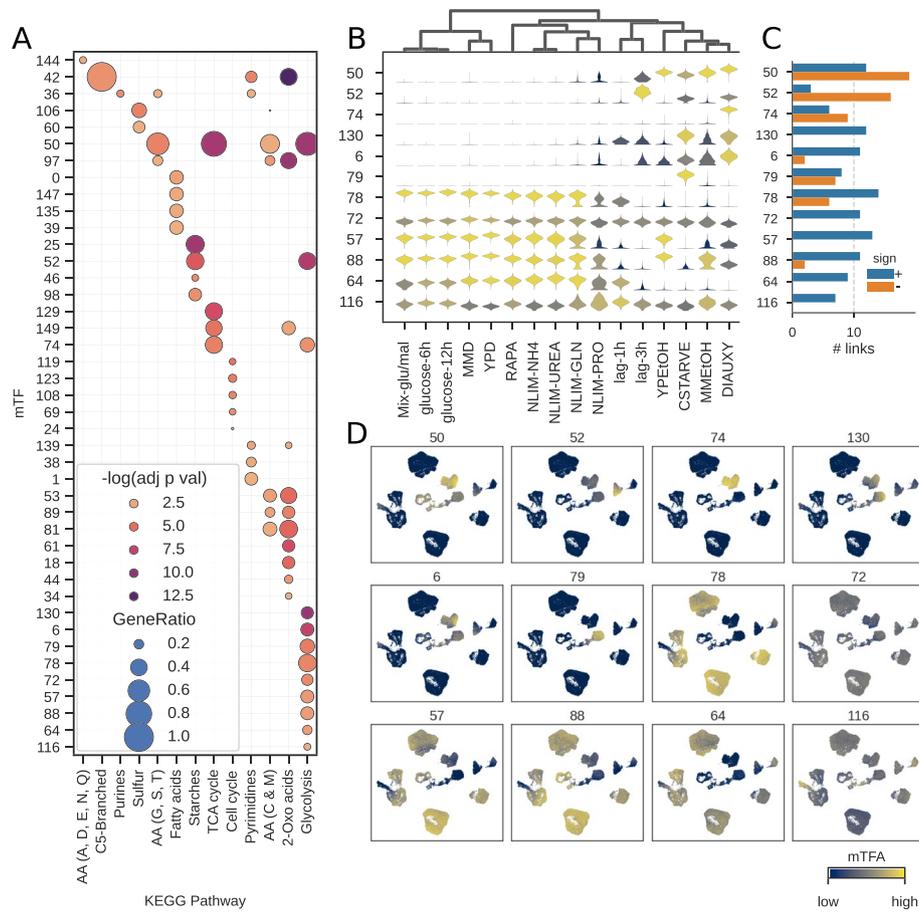


Fig. 4 Meta transcription factor (mTF) functional enrichment analysis in single cell yeast. mTFA estimated from hierarchical SupirFactor model trained in Fig. 3. mTFs are nodes in the SupirFactor model Φ latent layer and are numbered from 1 to 148. **A** Pathway enrichment of mTFs to selected core metabolic 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-dimensional UMAP projection. Cell metadata plotted in Additional file 1: Fig. S1

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 (see the “[Joint representation learning for context-specific constraints](#)” section). 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 (Additional file 1: Fig. 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 13,698 genes, connected through 492 mTFs in a latent mTF layer ($\xi^2 \geq 0$) (Additional file 1: Fig. S11C).

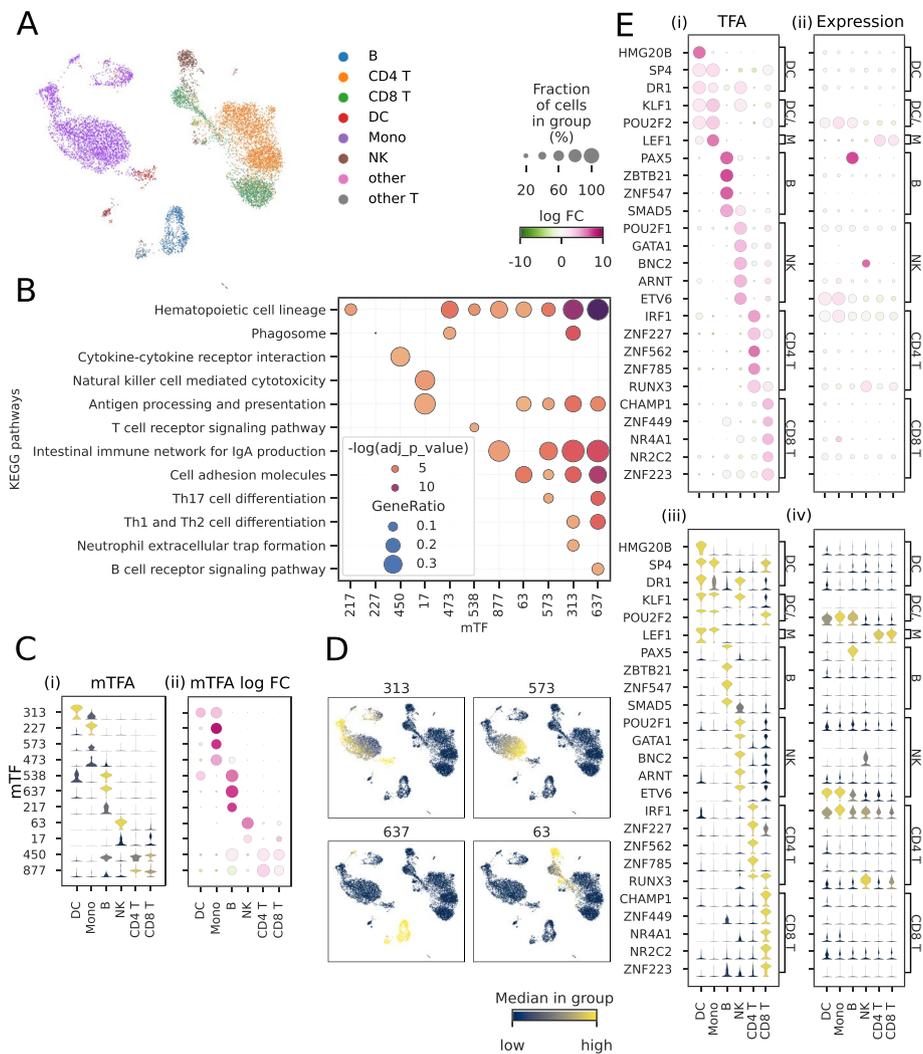


Fig. 5 Single cell PBMC TFA and mTF functional enrichment analysis. TFA and mTFA are estimated from a JRL SupirFactor model. mTFs are associated with latent features $\hat{\phi}$ and are numbered 0-881. **A** UMAP projection of the single cell PBMC dataset, labeled with cell type annotations. **B** Selected enriched terms and associated mTFs for cell-type specific KEGG pathway based on mTF target genes, $\xi^2 > 0.01$ (for mTF 227, phagosome pathway, adjusted $p = 0.0081$, gene ratio = 0.0309). **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

Eleven of these mTFs are linked to target genes which are functionally enriched for immune cell specific KEGG pathways (Fig. 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 (Fig. 5B-D); the target genes of this mTF are functionally enriched for B-cell receptor signaling (Fig. 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 (Fig. 5E). SupirFactor distinguishes activation between myeloid (monocytes and DCs) and lymphocytic (B, NK, and T cells) lineages (Fig. 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 CD8 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.

We apply the trained PBMC model to single-cell PBMCs collected from healthy and from COVID-19-infected individuals [58] (Additional file 1: Fig. S13). This trained model computes TFA for each cell, allowing determination of differentially active TFs for each cell type (Additional file 1: Fig. S13A). Immune cells from COVID-infected patients show an upregulation of core inflammation TF activities (e.g., the Activator protein-1 TF components FosB and JUN), and changes to differentiation and proliferation TF activities (e.g., RUNX3 and IRF5). We further identify several TFs which are not, in healthy cells, specific to any immune cell type, but are predicted to have cell type-specific changes to TF activity during infection (e.g., HMBOX1, ZNF24, ZNF691). Based on this analysis, we conclude that exploring condition-specific data with SupirFactor models can uncover upstream causal regulators that are both cell-type and condition specific.

Discussion

SupirFactor has been carefully benchmarked using both bulk RNA expression data sets and single cell RNA expression data sets. We rely on model organisms *Bacillus 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.

Additionally, we provide a novel metric for evaluating deep neural networks (DNN) architectures, ERV, specifically designed for the needs of GRN inference. 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 interpretation. ERV

also allows for post-training analysis of any gene expression data to determine which parts of the network is 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 interpretation, i.e., 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 using TF expression as the independent feature to model a GRN will 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 makes use of context-specific prior evidence to further restrict TF variance. And we find that we can extract functional enrichment based on annotated cell types 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, 59]. 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 issue by using a DNN architecture together with the transparency framework (ERV). We demonstrate this reuse by embedding new data and by contextually analyzing sub-networks and condition specific latent activation after the model has been trained, gaining insight not explicitly provided to the model before training.

Conclusion

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 (TFA) 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 gene regulatory network (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.

Additionally, in this work, we devise the explained relative variance (ERV) metric. ERV estimates the importance of each latent feature to each output feature, both directly and indirectly depending on the model layer in question. This metric has appealing properties for GRN inference, e.g., ERV is bounded and facilitates ranking regulatory

relationships and discarding non-predictive model weights when interpreting the model. As biological GRNs are sparse model selection must be part of the evaluation of any GRN inference algorithm and ERV facilitates this critical component. This model evaluation metric we propose is also useful for evaluating the contribution of intermediate DNN layers which are not explicitly defined as TFs and facilitate functional annotation and latent feature interpretation.

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 harmonize 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.

Methods

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.

Shallow SupirFactor

Starting from our model framework (see the “[The SupirFactor regulation model](#)” section), 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

$$\hat{\mathbf{x}} = h(\mathbf{x}) = (g \circ f)(\mathbf{x}) \quad (1)$$

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

$$\boldsymbol{\phi} = f(\mathbf{x}) = \sigma(\mathbf{W}\mathbf{x}) \quad (2)$$

and

$$\hat{\mathbf{x}} = g(\boldsymbol{\phi}) = \sigma(\boldsymbol{\Theta}\boldsymbol{\phi}) \quad (3)$$

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 “[Network component analysis \(NCA\)](#)” section, with the distinction that this formulation weighs \mathbf{W} instead of fitting ϕ with a static \mathbf{P} .

Θ and \mathbf{W} are the weighted connections of input features to output features in a single layer, with equation (3) corresponding to the shallow SupirFactor (Fig. 1C). ϕ is the inferred latent activity interpreted as TFA of expression mapped through f . We set $\mathbf{W} \in \mathbf{P}$, such that the sparsity structure of \mathbf{W} is identical to \mathbf{P} . This ensures that (I) $\phi \in \mathbb{R}^K$, with K TFs, constraining the informative genes influences to those in the prior \mathbf{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.

Hierarchical SupirFactor

In hierarchical SupirFactor (Fig. 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.

$$\hat{\mathbf{x}} = h(\mathbf{x}) = (g \circ s \circ f)(\mathbf{x}) \quad (4)$$

where

$$\hat{\phi} = s(\phi) = \sigma(\mathbf{\Pi}\phi) \quad (5)$$

and

$$\hat{\mathbf{x}} = g(\hat{\phi}) = \sigma(\Theta\hat{\phi}) \quad (6)$$

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

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) (Fig. 1E). This is implemented in SupirFactor by adding a biological context-specific constraint on the prior evidence. We define \mathbf{P}_C as prior evidence for C where C is a biological context, like a cell type, growth condition, or temporal group. Weights $\mathbf{W}_C \in \mathbf{P}_C$ are also context-specific and are mapped jointly through $\mathbf{\Pi}$ and Θ 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 \mathbf{W}_C are individually trained and may vary between contexts if \mathbf{P}_C is the same. Equation (4) then takes the form;

$$\hat{\mathbf{x}}_C = (g \circ s \circ f_C)(\mathbf{x}_C) \quad (7)$$

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

Fitting model

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

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

This is implemented as minimization by batch stochastic gradient descent with the Adam solver [60] and pytorch [61].

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

Model regularization**Parameter penalties**

Model weights are penalized, regularizing models to mitigate overfitting and to balance bias and variance [62]. We use a *weight decay* factor, corresponding to a ridge penalty (see [63], Section 6.7.6). The objective function to be minimized is then

$$\text{MSE} + \zeta \|\mathbf{W}\|_2 + \zeta \|\mathbf{\Theta}\|_2 + \zeta \|\mathbf{\Pi}\|_2 \quad (9)$$

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 $\zeta \in \{0, 10^{-10}, 10^{-9}, \dots, 10^{-1}\}$ [31, 64].

Dropout

Dropout is an additional regularization method [65] 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. Dropout is tested by cross validation as above on both the input and the latent TF layer, searching $p \in \{0, 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$.

Explained relative variance (ERV)**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 [35]. 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.

$$\xi_{i,k}^2 = 1 - \frac{\text{MSE}_i(\bar{h}(\mathbf{x})_{\phi})}{\text{MSE}_i(\bar{h}(\mathbf{x})_{\phi_{k \neq k}})} \quad (10)$$

\bar{h} is the trained model, $\text{MSE}_i(\bar{h}(\mathbf{x})_{\phi})$ is the MSE of the full, unperturbed model for gene i , and $\text{MSE}_i(\bar{h}(\mathbf{x})_{\phi_{k \neq k}})$ 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 $\xi_{i,k}^2$ for all k .

Ranking model parameters for models with multiple layers

To be able to interpret any parameter in the model we use the ERV concept (see the “[Model selection and model parameter ranking](#)” section). To determine the importance in any hidden layer L not directly connected to the output, we compute two ξ^2 matrices. ξ_L^2 and ξ_{L+1}^2 . 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 $\xi_{:,k,L}^2$ and $\xi_{:,m,L+1}^2$. 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 $\xi_{:,k,L}^2$. To eliminate weights in $\mathbf{\Pi}$ we threshold $\xi_{:,k,L+1}^2 > \epsilon_1$ and $\xi_{:,m,L}^2 > \epsilon_2$ and compute the ERV:

$$\xi_{m,k,\mathbf{\Pi}}^2 \equiv 1 - \frac{\sum_{i \in k \cap m} \text{MSE}_i(\bar{h}(\mathbf{x})_{\phi})}{\sum_{i \in k \cap m} \text{MSE}_{i,k,L}(\bar{h}(\mathbf{x})_{\phi_{k \neq k}})} \quad (11)$$

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 $\xi_{m,k,\mathbf{\Pi}}^2 \equiv 0$.

The classical GRN realization is interpreted as the indirect connections from $\xi_{i,k,L}^2$, 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.

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

$\xi_{i,k,L}^2 \leq \epsilon$ 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 (Fig. 1E). The model selection step is derived from ξ^2 where

$$\widehat{\xi}_{i,k}^2 \equiv \frac{\xi_{i,k}^2}{\overline{\xi}_i^2} \quad (12)$$

with $\overline{\xi}_i^2$ 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 $\widehat{\xi}_{i,k}^2 > \epsilon$. 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 $\widehat{\xi}_{i,k}^2$ 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 $\xi_k^2 \leq \epsilon$ 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 $\xi_{k,m,\Pi}^2 \leq 0$ the hidden layer weight is pruned.

Network component analysis (NCA)

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

$$Y = \phi \Theta \quad (13)$$

where $Y \in \mathbb{R}^{S,n}$ is gene expression with n genes and S samples, $\phi \in \mathbb{R}^{m,k}$ is the TFA with k TFs, and $\Theta \in \mathbb{R}^{k,n}$ is the regulatory effects linking genes to TFs. The unknown “true” Θ is the regulatory interaction between genes and TFs we want to find. ϕ is unknown given the assumption that the expression level of a transcription factor k does not correlate well with the activity of the protein [66]. Therefore, we need to solve for both Θ and ϕ , which forces us to convolve our estimation of regulatory effect and the TFA. To deconvolve and solve this, we impose a prior P with elements $\in \{0, 1\}$ as an initial guess to the structure of Θ and use that to solve for an initial estimate for ϕ .

$$\hat{\phi} = \arg \min_{\phi} \|Y - \phi P\| \quad (14)$$

This is solved by ordinary least squares. The estimated ϕ is then used to solve for Θ

$$\hat{\Theta} = \arg \min_{\Theta} \|Y - \hat{\phi} \Theta\| \quad (15)$$

This is interpreted as an estimate of TFA given a number of reporter genes defined by the prior 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.

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

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

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 .

Data acquisition

Bulk expression data

Bacillus subtilis bulk expression data for data set 1 (B1) [10] and data set 2 (B2) [67] and the known prior network [10] were used as previously described [41]. *Saccharomyces cerevisiae* bulk expression data for data set 1 [68] and data set 2 [69], and the known prior network [70] were used as previously described [41].

Single cell expression data

Saccharomyces cerevisiae single cell training data was assembled from [13] and [71] 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 (*MAT α /MAT α Δ ho::NatMX/ Δ ho::KanMX*) and a wild-type strain 2 (*MAT α /MAT α HO/ Δ ho::NatMX HAP1+::pACT1-Z3EV::NatMX/HAP1 *ura3 Δ 0/URA3 can1 Δ ::prSTE2-HIS5/CAN1 HIS3/his3 Δ 1 LYP1/lyp1 Δ 0*) were generated as previously described [13].*

Strains were grown overnight in rich media (YPD as previously described [13]) and then subcultured into 100 mL YPD or minimal media (MMD as previously described [13]) for 3 h. 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.

Yeast test GRN

Yeast network performance evaluation was performed using a previously described high-confidence gold-standard GRN [41, 68] containing 1403 regulatory edges connecting 993 genes to 98 TFs. The yeast prior knowledge network consists of 11,486 regulatory edges connecting 3912 genes to 152 TFs, and many of the edges are shared with the

gold-standard GRN, necessitating the hold-out for testing strategy used for performance evaluation.

An additional test GRN was constructed which had no edges from the prior knowledge network. This test GRN was obtained from the *Saccharomyces* Genome Database (yeastgenome.org) and consists of 29,253 regulatory edges connecting 6367 genes to 170 regulatory TFs. We further modify it by masking every regulatory edge which occurs in the prior knowledge network, excluding it from scoring. This yields a test GRN with 25,621 edges, none of which occur in the prior knowledge network or the gold-standard GRN, connecting 6367 genes to 170 regulatory TFs.

Yeast annotations

Cell cycle related yeast genes are annotated based on [72]. Ribosomal, ribosomal biogenesis, and induced environmental stress response genes are annotated based on [73].

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 (E, Y, W)), KEGG:00270 (AA (C & M)), KEGG:00920 (Sulfur), KEGG:00061 (Fatty acids), KEGG:00230 (Purines), and KEGG:00240 (Pyrimidines).

PBMC multi-ome dataset preprocessing

Paired PBMC scRNA-seq and scATAC-seq was downloaded from the 10x website [74]. This data was preprocessed using a previously published workflow [48]. In short, the RNA-seq data is preprocessed as detailed in the “Single cell pre-processing” section, with the additional filtering of cells with > 25000 or < 1000 counts and < 20% mitochondrial counts of total. For the ATACseq data, we used epiScanpy [75], filtering peaks in < 10 cells and cells with < 5000 or > $7 \cdot 10^4$ counts, and with a variability score < 0.515. Final data contains 10,411 cells, 21,601 genes, and 75,111 peaks.

Cell types are annotated using the reference PBMC dataset [59] passed to scanpy’s [76] inject label transfer function, resulting in 8 annotated celltypes (Fig. 5A).

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 50 kbp upstream of a gene transcription start site and 2 kbp 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 intersection with annotated regulatory regions for PBMC cell types (ENCODE Accession IDs ENCF776AJJ, ENCF497NXM, ENCF984SPH, ENCF079TQT, ENCF862ULW, ENCF504FDC, and ENCF905BHJ). The peak intensity (signalValue) was summed for all peaks annotated to each TF-gene pair to generate 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%.

PBMC COVID dataset preprocess

Healthy and COVID patient single cell data from [58] was fetched from [77]. Expression counts were preprocessed similarly as the PBMC data (see the “PBMC multi-ome dataset preprocessing” section).

Data preprocessing and model parameterization

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 and (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.

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 [78] which we call RobustMinScaler.

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

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

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

Data preprocessing and model parameterisation

Visualizations throughout this work, if not stated otherwise, were generated in Matplotlib [79] with some components done with seaborn [80] and scanpy [76].

Supplementary Information

The online version contains supplementary material available at <https://doi.org/10.1186/s13059-023-03134-1>.

Additional file 1. Supplementary information. This file contains all supplemental figures along with their descriptions.

Additional file 2. Peer review history.

Acknowledgements

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.

Review history

The review history is available as Additional file 2.

Peer review information

Tim Sands was the primary editor of this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Authors' contributions

Project conceptualization and design: A.T., with contributions from all other authors. Data acquisition: A.T., C.A.J. Model implementation: A.T., M.B. Analysis of results: A.T., M.B., C.A.J., with contributions from all other authors. Initial manuscript writing: A.T., M.B., C.A.J., R.B. Final manuscript editing: all authors.

Funding

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

Availability of data and materials

The most recent iteration of the open-source SupirFactor software, released under the permissive BSD 2-Clause "Simplified" License, is available at <https://gitlab.com/Xparx/supirfactor> [81]. The specific version of the software used in this publication is accessible on Zenodo through the following link: <https://zenodo.org/doi/10.5281/zenodo.10161545> [82]. The Zenodo repository also contains the trained models used in the manuscript for both single-cell yeast (scY) and peripheral blood mononuclear cell (PBMC) datasets. Additionally, the repository encompasses the inferred gene regulatory networks (GRNs) between transcription factors (TFs) and genes, and meta TFs and genes. Furthermore, it contains the TF activity (TFA) and mTFA for each model for the training data.

All publicly available and previously published data referenced and used in the manuscript are detailed in the "Data acquisition" section, with the repository on Zenodo containing the used versions of priors, gold standards, and bulk expression benchmark datasets.

Regarding the single-cell data, *S. cerevisiae* was downloaded from the NCBI GEO database. The training data originated from GSE144820 and GSE125162, while the validation data came from GSE218089.

PBMC data was obtained from https://support.10xgenomics.com/single-cell-multiome-atac-gex/datasets/1.0.0/pbmc_granulocyte_sorted_10k [74], and the additional PBMC COVID dataset was acquired from <https://ndownloader.figshare.com/files/27458837> [77].

Declarations

Ethics approval and consent to participate

Ethics approval were not needed for the study.

Competing interests

The authors declare that they have no competing interests.

Author details

¹Center for Developmental Genetics, New York University, New York, NY 10003, USA. ²Center For Genomics and Systems Biology, NYU, New York, NY 10008, USA. ³Department of Biology, NYU, New York, NY 10008, USA. ⁴Flatiron Institute, Center for Computational Biology, Simons Foundation, New York, NY 10010, USA. ⁵Courant Institute of Mathematical Sciences, Computer Science Department, New York University, New York, NY 10003, USA. ⁶Center For Data Science, NYU, New York, NY 10008, USA. ⁷Prescient Design, a Genentech accelerator, New York, NY 10010, USA. ⁸Mortimer B. Zuckerman Mind Brain Behavior Institute, Columbia University, New York, NY 10010, USA. ⁹Sars International Centre for Marine Molecular Biology, University of Bergen, Bergen, Norway. ¹⁰Department of Heart Disease, Haukeland University Hospital, Bergen, Norway. ¹¹Department of Neuro-Science, University of Wisconsin-Madison - Waisman Center, Madison, USA.

Received: 28 February 2023 Accepted: 30 November 2023

Published online: 18 January 2024

References

1. Casamassimi A, Ciccodicola A. Transcriptional Regulation: Molecules, Involved Mechanisms, and Misregulation. *Int J Mol Sci.* 2019;20(6). <https://doi.org/10.3390/ijms20061281>.
2. Chasman D, Fotuhi Siahpirani A, Roy S. Network-based approaches for analysis of complex biological systems. *Curr Opin Biotechnol.* 2016;39:157–66. <https://doi.org/10.1016/j.copbio.2016.04.007>.
3. Cho DY, Kim YA, Przytycka TM. Chapter 5: Network Biology Approach to Complex Diseases. *PLOS Comput Biol.* 2012;8(12):1–11. <https://doi.org/10.1371/journal.pcbi.1002820>.
4. Ritchie MD, Holzinger ER, Li R, Pendergrass SA, Kim D. Methods of integrating data to uncover genotype-phenotype interactions. *Nat Rev Genet.* 2015;16(2):85–97. <https://doi.org/10.1038/nrg3868>.
5. Goode D, Obier N, Vijayabaskar MS, Lie-A-Ling M, Lilly A, Hannah R, et al. Dynamic gene regulatory networks drive hematopoietic specification and differentiation. *Dev Cell.* 2016;36(5):572–87. <https://doi.org/10.1016/j.devcel.2016.01.024>.
6. Bernadskaya Y, Christiaen L. Transcriptional control of developmental cell behaviors. *Ann Rev Cell Dev Biol.* 2016;32(1):77–101. <https://doi.org/10.1146/annurev-cellbio-111315-125218>.
7. Latchman D. Transcription factors: an overview. *Int J Exp Pathol.* 1993;74(5):417–22.
8. Paraiso KD, Cho JS, Yong J, Cho KKY. Chapter Two - Early Xenopus gene regulatory programs, chromatin states, and the role of maternal transcription factors. In: Peter IS, editor. *Gene Regulatory Networks.* vol. 139 of *Current Topics in Developmental Biology.* Academic Press; 2020. p. 35–60. <https://doi.org/10.1016/bs.ctdb.2020.02.009>.
9. Schacht T, Oswald M, Eils R, Eichmüller SB, König R. Estimating the activity of transcription factors by the effect on their target genes. *Bioinformatics.* 2014;30(17):i401–7. <https://doi.org/10.1093/bioinformatics/btu446>.
10. Arrieta-Ortiz ML, Hafemeister C, Bate AR, Chu T, Greenfield A, Shuster B, et al. An experimentally supported model of the *Bacillus subtilis* global transcriptional regulatory network. *Mol Syst Biol.* 2015;11(11). <https://doi.org/10.15252/msb.20156236>.
11. Shi Q, Zhang C, Guo W, Zeng T, Lu L, Jiang Z, et al. Local network component analysis for quantifying transcription factor activities. *Methods.* 2017;124:25–35. <https://doi.org/10.1016/j.ymeth.2017.06.018>. Integrative Analysis of Omics Data.
12. Kao KC, Yang YL, Boscolo R, Sabatti C, Roychowdhury V, Liao JC. Transcriptome-based determination of multiple transcription regulator activities in *Escherichia coli* by using network component analysis. *Proc Natl Acad Sci USA.* 2004;101(2):641–6. <https://doi.org/10.1073/pnas.0305287101>.
13. Jackson CA, Castro DM, Saldi GA, Bonneau R, Gresham D. Gene regulatory network reconstruction using single-cell RNA sequencing of barcoded genotypes in diverse environments. *eLife.* 2020;9:e51254. <https://doi.org/10.7554/eLife.51254>.
14. Castro DM, de Veaux NR, Miraldi ER, Bonneau R. Multi-study inference of regulatory networks for more accurate models of gene regulation. *PLOS Comput Biol.* 2019;15(1):1–22. <https://doi.org/10.1371/journal.pcbi.1006591>.
15. Kamimoto K, Hoffmann CM, Morris SA. CellOracle: dissecting cell identity via network inference and in silico gene perturbation. *bioRxiv.* 2020. <https://doi.org/10.1101/2020.02.17.947416>.
16. Chen Y, Li Y, Narayan R, Subramanian A, Xie X. Gene expression inference with deep learning. *Bioinformatics.* 2016;32(12):1832–9. <https://doi.org/10.1093/bioinformatics/btw074>.
17. Chen L, Cai C, Chen V, Lu X. Learning a hierarchical representation of the yeast transcriptomic machinery using an autoencoder model. *BMC Bioinformatics.* 2016;17(1):S9. <https://doi.org/10.1186/s12859-015-0852-1>.
18. Chitto D, Sadowski P, Baldi P. 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.* New York: Association for Computing Machinery; 2014. p. 533–40. <https://doi.org/10.1145/2649387.2649442>.
19. Dwivedi SK, Tjärnberg A, Tegnér J, Gustafsson M. Deriving disease modules from the compressed transcriptional space embedded in a deep autoencoder. *Nat Commun.* 2020;11(1):856. <https://doi.org/10.1038/s41467-020-14666-6>.
20. Chen HH, Chiu YC, Zhang T, Zhang S, Huang Y, Chen Y. GSAE: an autoencoder with embedded gene-set nodes for genomics functional characterization. *BMC Syst Biol.* 2018;12(8):142. <https://doi.org/10.1186/s12918-018-0642-2>.
21. Yuan Y, Bar-Joseph Z. Deep learning for inferring gene relationships from single-cell expression data. *Proc Natl Acad Sci.* 2019;116(52):27151–8. <https://doi.org/10.1073/pnas.1911536116>.
22. Lopez R, Regier J, Cole MB, Jordan MI, Yosef N. Deep generative modeling for single-cell transcriptomics. *Nat Methods.* 2018;15(12):1053–8. <https://doi.org/10.1038/s41592-018-0229-2>.
23. Novakovskiy G, Dexter N, Libbrecht MW, Wasserman WW, Mostafavi S. Obtaining genetics insights from deep learning via explainable artificial intelligence. *Nat Rev Genet.* 2022. <https://doi.org/10.1038/s41576-022-00532-2>.
24. Covert I, Lundberg S, Lee SI. Explaining by removing: a unified framework for model explanation. *arXiv.* 2020. <https://doi.org/10.48550/ARXIV.2011.14878>.
25. Sung AH. Ranking importance of input parameters of neural networks. *Exp Syst Appl.* 1998;15(3):405–11. [https://doi.org/10.1016/S0957-4174\(98\)00041-4](https://doi.org/10.1016/S0957-4174(98)00041-4).
26. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: visual explanations from deep networks via gradient-based localization. *Int J Comput Vis.* 2019;128(2):336–59. <https://doi.org/10.1007/s11263-019-01228-7>.
27. Di Bernardo D, Gardner TS, Collins JJ. Robust identification of large genetic networks. *Pac Symp Biocomput Pacific Symp Biocomput.* 2004;497:486–97.
28. Bonneau R, Reiss DJ, Shannon P, Facciotti M, Hood L, Baliga NS, et al. The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome Biol.* 2006;7(5):R36. <https://doi.org/10.1186/gb-2006-7-5-r36>.

29. Bonneau R. Learning biological networks: from modules to dynamics. *Nat Chem Biol.* 2008;4(11):658–64. <https://doi.org/10.1038/nchembio.122>.
30. Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data mining, inference and prediction. 2nd ed. Springer; 2009. <https://hastie.su.domains/ElemStatLearn/>.
31. Tjärnberg A, Nordling TEM, Studham M, Sonnhammer ELL. Optimal Sparsity Criteria for Network Inference. *J Comput Biol.* 2013;20(5):398–408. <https://doi.org/10.1089/cmb.2012.0268>.
32. Wonkap SK, Butler G. BENIN: Biologically enhanced network inference. *J Bioinform Comput Biol.* 2020;18(03):2040007. <https://doi.org/10.1142/S0219720020400077>.
33. Madar A, Greenfield A, Vanden-Eijnden E, Bonneau R. DREAM3: network inference using dynamic context likelihood of relatedness and the inferelator. *PLoS ONE.* 2010;5(3):e9803–e9803. <https://doi.org/10.1371/journal.pone.0009803>.
34. Huynh-Thu VA, Irrthum A, Wehenkel L, Geurts P. Inferring regulatory networks from expression data using tree-based methods. *PLoS ONE.* 2010;5(9):1–10. <https://doi.org/10.1371/journal.pone.0012776>.
35. Magnusson R, Gustafsson M. LiPLike: towards gene regulatory network predictions of high certainty. *Bioinformatics.* 2020;36(8):2522–9. <https://doi.org/10.1093/bioinformatics/btz950>.
36. Fortelny N, Bock C. Knowledge-primed neural networks enable biologically interpretable deep learning on single-cell sequencing data. *Genome Biol.* 2020;21(1):190. <https://doi.org/10.1186/s13059-020-02100-5>.
37. Aibar S, González-Blas CB, Moerman T, Huynh-Thu VA, Imrichova H, Hulselmans G, et al. SCENIC: single-cell regulatory network inference and clustering. *Nat Methods.* 2017;14(11):1083–6. <https://doi.org/10.1038/nmeth.4463>.
38. Bauckhage C, Ojeda C, Schücker J, Sifa R, Wrobel S. Informed machine learning through functional composition. In: *Lernen, Wissen, Daten, Analysen.* 2018. <https://api.semanticscholar.org/CorpusID:52162764>.
39. Buenrostro JD, Giresi PG, Zaba LC, Chang HY, Greenleaf WJ. Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, DNA-binding proteins and nucleosome position. *Nat Methods.* 2013;10:1213–8. <https://doi.org/10.1038/nmeth.2688>.
40. Lee TI, Rinaldi NJ, Robert F, Odom DT, Bar-Joseph Z, Gerber GK, et al. Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science.* 2002;298(5594):799–804. <https://doi.org/10.1126/science.1075090>.
41. Skok Gibbs C, Jackson CA, Saldi GA, Tjärnberg A, Shah A, Watters A, et al. High-performance single-cell gene regulatory network inference at scale: the Inferelator 3.0. *Bioinformatics.* 2022;38(9):2519–28. <https://doi.org/10.1093/bioinformatics/btac117>.
42. Teixeira MC, Viana R, Palma M, Oliveira J, Galocha M, Mota MN, et al. YEASTRACT+: a portal for the exploitation of global transcription regulation and metabolic model data in yeast biotechnology and pathogenesis. *Nucleic Acids Res.* 2023;51(D1):D785–91. <https://doi.org/10.1093/nar/gkac1041>.
43. Anderson-Sprecher R. Model comparisons and R^2 . *Am Stat.* 1994;48(2):113–7. <https://doi.org/10.1080/00031305.1994.10476036>.
44. Goodfellow I, Bengio Y, Courville A. *Deep Learning.* MIT Press; 2016. <http://www.deeplearningbook.org>.
45. Bähler J. Cell-cycle control of gene expression in budding and fission yeast. *Annu Rev Genet.* 2005;39:69–94.
46. Zaman S, Lippman SI, Zhao X, Broach JR. How *Saccharomyces* responds to nutrients. *Annu Rev Genet.* 2008;42:27–81.
47. Kanehisa M, Goto S. KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.* 2000;28(1):27–30.
48. Hao Y, Hao S, Andersen-Nissen E, Mauck WM, Zheng S, Butler A, et al. Integrated analysis of multimodal single-cell data. *Cell.* 2021;184(13):3573–3587.e29. <https://doi.org/10.1016/j.cell.2021.04.048>.
49. Traag VA, Waltman L, van Eck NJ. From Louvain to Leiden: guaranteeing well-connected communities. *Sci Rep.* 2019;9(1):5233. <https://doi.org/10.1038/s41598-019-41695-z>.
50. Sangaletti S, Tripodo C, Chiodoni C, Guarotta C, Cappetti B, Casalini P, et al. Neutrophil extracellular traps mediate transfer of cytoplasmic neutrophil antigens to myeloid dendritic cells toward ANCA induction and associated autoimmunity. *Blood.* 2012;120(15):3007–18. <https://doi.org/10.1182/blood-2012-03-416156>.
51. Luo Q, Ma X, Wahl SM, Bieker JJ, Crossley M, Montaner LJ. Activation and repression of interleukin-12 p40 transcription by erythroid Kruppel-like factor in macrophages*. *J Biol Chem.* 2004;279(18):18451–6. <https://doi.org/10.1074/jbc.M400320200>.
52. Cobaleda C, Schebesta A, Delogu A, Busslinger M. Pax5: the guardian of B cell identity and function. *Nat Immunol.* 2007;8(5):463–70. <https://doi.org/10.1038/ni1454>.
53. Malhotra N, Kang J. SMAD regulatory networks construct a balanced immune system. *Immunology.* 2013;139(1):1–10. <https://doi.org/10.1111/imm.12076>.
54. Trikha P, Moseman JE, Thakkar A, Campbell AR, Elmas E, Foltz JA, et al. Defining the AHR-regulated transcriptome in NK cells reveals gene expression programs relevant to development and function. *Blood Adv.* 2021;5(22):4605–18. <https://doi.org/10.1182/bloodadvances.2021004533>.
55. Kano Si, Sato K, Morishita Y, Vollstedt S, Kim S, Bishop K, et al. The contribution of transcription factor IRF1 to the interferon- γ -interleukin 12 signaling axis and TH1 versus TH-17 differentiation of CD4+ T cells. *Nat Immunol.* 2008;9(1):34–41. <https://doi.org/10.1038/ni1538>.
56. Korinfskaya S, Parameswaran S, Weirauch MT, Barski A. Runx transcription factors in T cells-what is beyond thymic development? *Front Immunol.* 2021;12. <https://doi.org/10.3389/fimmu.2021.701924>.
57. Liu X, Wang Y, Lu H, Li J, Yan X, Xiao M, et al. Genome-wide analysis identifies NR4A1 as a key mediator of T cell dysfunction. *Nature.* 2019;567(7749):525–9. <https://doi.org/10.1038/s41586-019-0979-8>.
58. Arunachalam PS, Wimmers F, Mok CKP, Perera RAPM, Scott M, Hagan T, et al. Systems biological assessment of immunity to mild versus severe COVID-19 infection in humans. *Science.* 2020;369(6508):1210–20. <https://doi.org/10.1126/science.abc6261>.
59. Stuart T, Butler A, Hoffman P, Hafemeister C, Papalexi E, Mauck WM III, et al. Comprehensive integration of single-cell data. *Cell.* 2019;177(7):1888–1902.e21. <https://doi.org/10.1016/j.cell.2019.05.031>.
60. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. 2014.
61. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: an imperative style, high-performance deep learning library. In: Wallach H, Larochelle H, Beygelzimer A, d Alché-Buc F, Fox E, Garnett R, editors. *Advances in*

- Neural Information Processing Systems. vol 32. Curran Associates, Inc.; 2019. p. 8024–35. <https://openreview.net/forum?id=BJJsrmfCZ>.
62. Hilt DE, Seegrift DW, United States. Forest Services. Northeastern Forest Experiment Station (Radnor. Pa.). Ridge, a computer program for calculating ridge regression estimates. vol. no.236. Upper Darby, Pa, Dept. of Agriculture, Forest Service, Northeastern Forest Experiment Station, 1977; 1977. <https://www.biodiversitylibrary.org/bibliography/68934>.
 63. Beck JV, Arnold KJ. Parameter estimation in engineering and science. Probability and Statistics Series. Wiley; 1977. https://books.google.com/books?id=_qAYgYN87UQC.
 64. Golub GH, Heath M, Wahba G. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*. 1979;21(2):215–23. <https://doi.org/10.2307/1268518>.
 65. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res*. 2014;15(56):1929–58.
 66. Vogel C, Marcotte EM. Insights into the regulation of protein abundance from proteomic and transcriptomic analyses. *Nat Rev Genet*. 2012;13:227–32. <https://doi.org/10.1038/nrg3185>.
 67. Nicolas P, Mäder U, Dervyn E, Rochat T, Leduc A, Pigeonneau N, et al. Condition-dependent transcriptome reveals high-level regulatory architecture in *Bacillus subtilis*. *Science*. 2012;335(6072):1103–6.
 68. Tchourine K, Vogel C, Bonneau R. Condition-specific modeling of biophysical parameters advances inference of regulatory networks. *Cell Rep*. 2018;23(2):376–88. <https://doi.org/10.1016/j.celrep.2018.03.048>.
 69. Hackett SR, Baltz EA, Coram M, Wranik BJ, Kim G, Baker A, et al. Learning causal networks using inducible transcription factors and transcriptome-wide time series. *Mol Syst Biol*. 2020;16(3):e9174. <https://doi.org/10.15252/msb.20199174>.
 70. Monteiro PT, Oliveira J, Pais P, Antunes M, Palma M, Cavalheiro M, et al. YEASTRACT+: a portal for cross-species comparative genomics of transcription regulation in yeasts. *Nucleic Acids Res*. 2020;48(D1):D642–9.
 71. Jariani A, Vermeersch L, Cerulus B, Perez-Samper G, Voordeckers K, Van Brussel T, et al. A new protocol for single-cell RNA-seq reveals stochastic gene expression during lag phase in budding yeast. *eLife*. 2020;9:e55320.
 72. Spellman PT, Sherlock G, Zhang MQ, Iyer VR, Anders K, Eisen MB, et al. Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol Biol Cell*. 1998;9(12):3273–97.
 73. Gasch AP, Yu FB, Hose J, Escalante LE, Place M, Bacher R, et al. Single-cell RNA sequencing reveals intrinsic and extrinsic regulatory heterogeneity in yeast responding to stress. *PLoS Biol*. 2017;15(12): e2004050.
 74. genomics X. Paired PBMC scRNA-seq and scATAC-seq. 2022. https://support.10xgenomics.com/single-cell-multi-ome-atac-gex/datasets/1.0.0/pbmc_granulocyte_sorted_10k. Accessed 15 Dec 2022.
 75. Danese A, Richter ML, Chaichoopu K, Fischer DS, Theis FJ, Colomé-Tatché M. EpiScanpy: integrated single-cell epigenomic analysis. *Nat Commun*. 2021;12(1):5228. <https://doi.org/10.1038/s41467-021-25131-3>.
 76. Wolf FA, Angerer P, Theis FJ. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol*. 2018;19(1):15. <https://doi.org/10.1186/s13059-017-1382-0>.
 77. Arunachalam PS, Wimmers F, Mok CKP, Perera RAPM, Scott M, Hagan T, et al. Healy and Covid patient single cell data. 2023. <https://ndownloader.figshare.com/files/27458837>. Accessed 11 Jan 2023.
 78. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: machine learning in Python. *J Mach Learn Res*. 2011;12:2825–30.
 79. Hunter JD. Matplotlib: a 2D graphics environment. *Comput Sci Eng*. 2007;9(3):90–5. <https://doi.org/10.1109/MCSE.2007.55>.
 80. Waskom ML. seaborn: statistical data visualization. *J Open Source Softw*. 2021;6(60):3021. <https://doi.org/10.21105/joss.03021>.
 81. Tjärnberg A. StrUcture Primed Inference of Regulation using latent Factor ACTivity. 2023. <https://gitlab.com/Xparx/supirfactor>. Accessed 31 Jan 2023.
 82. Tjärnberg A, Beheler-Amass M, Jackson C, Christiaen L, Gresham D, Bonneau R. SupirFactor published models and generated data. Zenodo. 2023. <https://doi.org/10.5281/zenodo.10161546>.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.