Algorithmic Enhancements to Causal Inference Problems

by

Bingran Shen

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

New York University

September, 2024

Professor Dennis Shasha

Dedication

To my wife Dr. Yuyan Wang, with affection.

Acknowledgements

First of all, wish to express my most sincere gratitude and appreciation to my PhD supervisor, Dr. Dennis Shasha for his constant support and guidance through the whole doctoral program. I would also like to give special thanks to Dr. Gloria Coruzzi who, while not being my thesis advisor, was a spiritual advisor and the source of so many of our research problems. Next, I'd like to thank the other members of my committee Dr. Rajesh Raganath, Dr. Manpreet Katari and Dr. Tim Jeffers. Finally, I would like to acknowledge the suggestions and help of Dr. Jacopo Cirrone and Dr. Ji Huang.

Last but certainly not the least, I want to thank my parents, Huiling An and Xiaofei Shen, my wife Dr. Yuyan Wang for their ever lasting love and support.

Abstract

Finding causal relationships among biological entities is a core goal of systems biology. In molecular biology, causal relationships are often encoded as simple graphs called gene regulatory networks (GRNs). The nodes of a GRN are genes and its directed edges represent positive or negative influences of a regulatory gene and its targets.

To infer a network, researchers often develop a machine learning model and then evaluate the model based on their match with experimentally verified "gold standard" edges. The hoped-for result of such a model is a network that may extend the gold standard edges.

Chapter 2 introduces *EnsInfer*, a novel approach to infer GRNs from RNA-seq data. Following a systematic evaluation of the performance of various existing base network inference methods and ensemble techniques across different datasets, we found that: (i) base inference methods exhibit varying performances across datasets, with a method performing poorly on one dataset potentially excelling on another; (ii) a non-homogeneous ensemble method, specifically a Naive Bayes classifier, achieves overall performance on par with or better than the best individual method or other ensemble approaches; (iii) integrating all methods satisfying a statistical normality test on training data yields optimal results. Motivated by these insights, *EnsInfer* combines state-of-the-art inference algorithms using a Naive Bayes classifier. *EnsInfer* outperforms individual methods, providing a robust and flexible framework for integrating diverse data types and seamlessly incorporating new inference algorithms. Notably, by treating predictions from different algorithms as priors for each network edge, *EnsInfer* readily accommodates heterogeneous data sources (e.g.,

bulk and single-cell RNA-seq) and facilitates the integration of emerging inference techniques.

Our second contribution proposes a simple answer to what might seem to be a naive question: are GRNs useful models of causality? Since networks are a form of visual representation, one can compare their utility with architectural or machine blueprints. Blueprints are clearly useful because they give precise guidance to builders in construction. If the primary role of GRNs is to characterize causality, then such networks should be good tools for prediction because prediction is the actionable benefit of knowing causality.

Chapter 3 explores this question by asking how good GRNs are at prediction. We compare nonlinear machine learning models inferred from time-series data across four species against "gold standard" regulatory edges from previous experiments. The nonlinear models achieve better predictive performance, with more reductions in root mean square error (RMSE) compared to models based solely on the gold standard edges. This finding suggested that conventional GRNs fail to fully characterize causality. To address this limitation, we introduced a *bipartite network* representation, where nodes represent either genes and models, to better capture the synergistic regulatory effects of multiple transcription factors on target genes.

Further, we propose that the causality in regulation should focus on four key goals and we provide algorithms for all goals: (i) predictive accuracy, (ii) parsimonious enumeration of predictive regulatory genes for each target, (iii) identification of disjoint sets of predictive regulatory genes with roughly equal accuracy for each target, and (iv) construction of a *bipartite network* representation of causality. Our *bipartite network* framework offers an actionable and interpretable paradigm for investigating causal gene regulation. In addition to modeling transcriptional causality, the *bipartite network* framework may have far broader applicability, providing a powerful approach to causality research across diverse domains.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

Biological systems are inherently intricate, exhibiting emergent properties such as nonlinearity, feedback loops, and adaptation, arising from the intricate interplay between system components and their environment [Artime and De Domenico 2022]. Gene regulatory networks (GRNs) constitute one representation of the interrelationships among genes. These networks, particularly those focusing on transcriptomics, attempt to characterize the intricate web of interactions between transcription factors (TFs) and their target genes [Vijesh et al. 2013; Emmert-Streib et al. 2014]. GRNs are also abstract mathematical representations, depicting the relationships between biological entities as directed graphs, where vertices represent genes, mRNAs, or TFs, and edges denote regulatory interactions. The network's topological structure provides potentially valuable insights into the temporal dynamics and key regulatory hierarchies governing biological processes at the cellular level.

A crucial challenge for GRNs is inferring the underlying network structure from experimental data, a task known as GRN inference. GRN inference seeks to reverse-engineer the network topology from high-throughput omics data, such as gene expression profiles, protein-DNA binding affinity measurements, or chromatin accessibility assays [Madhamshettiwar et al. 2012]. By integrating these diverse data sources, researchers aim to decipher the complex web of interactions between regulatory elements and their target genes. However, this endeavor faces numerous challenges, including the inherent noise and variability present in biological data, the high dimensionality of the search space, and the potential for incomplete or indirect measurements [Liang et al. 1998].

GRN inference process can be broadly categorized into two groups of approaches: steady-state and time-series gene expression methods [Liang et al. 1998]. While steady-state methods rely on gene expression data from a single time point, time-series data offer a more informative perspective by capturing the dynamic nature of gene expression over time. Time-series transcriptomics involves measuring gene expression at multiple time points, providing a comprehensive view of the temporal dynamics and regulatory mechanisms governing gene regulation [Bar-Joseph et al. 2012]. Time-series transcriptomics data have the potential to unravel the intricate temporal patterns and causal relationships underlying gene regulation. Although generating high-quality time-series data can be experimentally challenging and resource-intensive, often requiring careful experimental design and sophisticated data normalization and preprocessing techniques [Spies and Ciaudo 2015], the wealth of information provided by time-series data has proven invaluable in advancing our understanding of complex biological processes, such as development, stress response, and disease progression.

GRN Inference methods based on time-series data aim to assign weights or scores to putative interactions between genes, reconstructing GRNs that best reflect the observed expression profiles. These weights quantify the strength and directionality of regulatory interactions, enabling the identification of key regulators and their target genes [Greenfield et al. 2010]. Over the years, various inference methods have been developed, which can be broadly categorized as model-free or model-based approaches. Model-free methods, such as correlation measures [Kim 2015] and mutual information [Qiu et al. 2018], rely on statistical dependencies between gene expression profiles to infer regulatory relationships. On the other hand, model-based approaches employ explicit mathematical models to describe the dynamics of gene regulation. These include ordinary differential equations [Bonneau et al. 2006], Bayesian reasoning [Kharchenko et al. 2014], and decision-tree-based models [Huynh-Thu et al. 2010]. In real-world applications, it has been demonstrated repeatedly that no single inference method can universally fit different inference tasks [Androulakis et al. 2007; Delgado and Gómez-Vela 2019; Huynh-Thu and Sanguinetti 2019; Pratapa et al. 2020]. The performance of these methods can vary depending on factors such as the complexity of the underlying network, the quality and quantity of the available data, and the specific biological context. This lack of a unanimous solution to the inference problem has prompted researchers to explore more sophisticated and adaptive approaches.

Therefore, the absence of a one-size-fits-all solution to the GRN inference challenge has become a major motivation for this thesis. It underscores the need for a more comprehensive and high-level approach that can leverage the strengths of multiple inference methods while mitigating their individual limitations. By embracing the diversity of available techniques and exploring novel ways to integrate them, we aim to provide algorithmic enhancements to help solve some important network causal inference problems.

1.2 EnsInfer: Non-Homogeneous Ensemble Network Inference

In Chapter 2, we propose a novel non-homogeneous stacking ensemble learning framework that harnesses the diversity of existing GRN inference methods. Through our research, we demonstrate the following key findings:

(i) Base network inference methods exhibit varying performance across different datasets, indicating that a method that performs poorly on one dataset may excel on another. This variability highlights the inherent challenges posed by the complexity and heterogeneity of biological network systems, underscoring the need for a more flexible and adaptive approach.

(ii) The non-homogeneous ensemble method that we propose, based on a Naive Bayes classifier, leads to overall performance that equals or surpasses the best single base method or any other ensemble method evaluated. This finding emphasizes the potential of ensemble learning techniques to leverage the strengths of multiple inference algorithms while mitigating their individual limitations.

(iii) To achieve these good results, the ensemble method should integrate all base methods that satisfy a statistical test of normality on the training data. This selective integration ensures that the ensemble leverages only those base methods that exhibit a consistent and well-behaved performance, thereby enhancing the robustness and reliability of the overall inference process.

Our proposed ensemble learning framework, dubbed *EnsInfer*, can be easily constructed by leveraging existing network inference methods through the following steps:

1. **Base Method Execution**: Execute a diverse set of individual base network inference algorithms on the data of interest. This step allows us to harness the unique strengths and perspectives offered by different inference approaches, ranging from correlation-based methods to advanced machine-learning techniques.

2. **Feeding the Ensemble Model**: Utilize the outputs obtained from the individual inference algorithms as input features for a second-level ensemble learning algorithm.

3. **Ensemble Learning**: Employ a robust ensemble learning algorithm, such as a Naive Bayes classifier or a random forest model, to combine the features extracted from the base methods. This integration step leverages the collective wisdom of multiple inference approaches, enabling the ensemble to capture complex patterns and dependencies that may be overlooked by individual methods.

4. **Network Prediction**: The ensemble learning algorithm trained on the integrated features predicts the regulatory edges of the GRN.

By following this modular and extensible framework, *EnsInfer* can seamlessly incorporate new or improved inference algorithms as they become available, continuously enhancing its predictive capabilities. Moreover, the ensemble learning approach enables the flexible integration of additional data sources, such as prior biological knowledge or complementary omics data, further refining the inferred regulatory networks.

Through benchmarking on real-world experimental data, we demonstrate the significant benefits offered by the proposed

1.3 Bipartite Networks: Towards a Better Representation of Causality

The GRN abstraction oversimplifies the underlying biological complexities, failing to capture the combinatorial synergistic effects that causal factors can exert [Reiter et al. 2017].

In Chapter 3, we compare the prediction quality based on "gold standard" regulatory edges from previous experimental work with non-linear models inferred from time-series data across four different species. Our findings suggest that the non-linear machine learning models, trained on time-series data, consistently outperform models based solely on the gold standard edges, with improvements ranging from 5.3% to 25.3% in terms of reduction in root mean squared error (RMSE). These results indicate that networks derived from gold standard edges alone fail to adequately capture the complexities of causal relationships governing gene regulation. Consequently, we propose that causality research should focus on the following four goals:

(i) Achieving high predictive accuracy by leveraging the power of machine learning models trained on expression data.

(ii) Identifying a parsimonious set of predictive regulatory genes for each target gene, reducing redundancy and improving interpretability.

(iii) Uncovering disjoint sets of predictive regulatory genes for each target gene, each offering comparable predictive accuracy, thereby capturing the potential existence of multiple redundant regulatory mechanisms.

(iv) Constructing a bipartite network representation of causality, with nodes representing genes and predictive models, to provide a holistic and multifaceted view of regulatory relationships.

Chapter 3 describes the details of our proposed algorithms and methodologies designed to address these four goals, paving the way for a more comprehensive and actionable understanding of causality in transcriptional networks and other applications.

1.4 Overall Contribution

The studies presented in this thesis demonstrate that both ensemble methods and bipartite network representations offer significant complementary benefits to traditional network inference workflows. Specifically, our proposed ensemble learning framework *EnsInfer* presents superior performance compared to individual network inference methods. By intelligently integrating diverse approaches, *EnsInfer* leverages the collective strengths of multiple algorithms while mitigating their individual limitations. This synergistic combination results in more accurate and robust GRN predictions, underscoring the value of embracing ensemble learning techniques in the field of systems biology.

Complementing the ensemble learning approach, the introduction of bipartite networks offers a robust and versatile framework for representing and analyzing complex causal relationships in biological systems. These contributions pave the way for new discoveries and practical applications in areas such as disease biology, drug development, and synthetic biology, ultimately advancing our understanding of the complex orchestra of life.

2 | Ensemble of Network Inference

2.1 Introduction

Network Inference

A gene regulatory network (GRN) consists of molecular regulators (including deoxyribonucleic acid (DNA) segments, messenger ribonucleic acid (RNA), and transcription factors) in a cell and the causal links between regulators and gene targets. Causality here means that the regulator influences the RNA expression of the gene target. Network inference is the problem of identifying such causal links. In machine learning terms, since the set of regulator genes and target genes are given, the network inference problem can be viewed as a binary classification task to determine whether or not a potential regulatory edge between any pair of regulator and target gene exists.

Because network inference facilitates the understanding of the biological systems at the level of molecular interactions, it potentially enables the designed repression or enhancement of groups of molecules. This has applications ranging from drug design and medical treatment to the reduced use of fertilizer in agriculture. Accurate network inference and functional validation is an ongoing challenge for systems biology. Over the past two decade, numerous gene regulatory network inference technologies have been proposed to tackle this problem from different perspectives ([Bonneau et al. 2006; Huynh-Thu et al. 2010; Matsumoto et al. 2017; Zheng et al. 2019; Shu et al. 2021; Zhao et al. 2021]).

Individual Methods Feed an Ensemble Method

Pratapa et al., ([Pratapa et al. 2020]) presented a framework called BEELINE to evaluate state-of-the-art network inference algorithms. The vast majority of the inference algorithms including the ones we are going to incorporate into the ensemble approach can be roughly categorized into three types below:

1. Pairwise correlation models make use of various kinds of correlations between a target gene's expression and potentially causal transcription factor expressions. PPCOR ([Kim 2015]) computes the partial and semi-partial correlation coefficients for each gene pair. LEAP ([Specht and Li 2017]) calculates the Pearson coefficient of each gene pair in a time-series background that considers a time-delay in regulatory response. PIDC ([Chan et al. 2017]) looks at the distribution of the gene expression and calculates the gene pair-wise mutual information between distributions. SCRIBE ([Qiu et al. 2018]) also looks at mutual information between gene expression distributions, and, like LEAP, considers time-lagged correlation in time-series data. Finally, there is a correlation on any set of steady-state data.

2. Tree-based models use random forests (or their close variants) to predict the gene expression of each target gene based on the expression of regulator genes (transcription factors). Such models then use feature importance to determine the weight of each regulator-target interaction. High weights correspond to regulatory edges. Examples include GENIE3 ([Huynh-Thu et al. 2010]), a faster alternative GRNBoost2 ([Moerman et al. 2019]), and the inference method OutPredict ([Cirrone et al. 2020]). OutPredict also takes prior information (e.g., binding data) into account during training and testing.

3. Ordinary differential equation (ODE)-based regression approaches model the target gene expression as a dependent on the time derivative of the expression of regulatory genes. Inferelator ([Bonneau et al. 2006]) is a regularized regression model that focuses on feature

selection. Its latest iteration, Inferelator 3.0 ([Gibbs et al. 2021]), makes use of single-cell data to learn regulatory networks. SCODE ([Matsumoto et al. 2017]) is a direct application of fast ODE-based regression. SINCERITIES ([Papili Gao et al. 2018]) utilizes Kolmogorov-Smirnov test-based ridge regression. GRISLI ([Aubin-Frankowski and Vert 2020]) is an ODE solver that accounts for gene expression velocity.

The BEELINE benchmark of 12 different inference algorithms showed that while some algorithms generally perform better than others, there is no definitive best solution that can be applied to all datasets. Our approach complements theirs: in addition to studying the performance of individual algorithms (including some promising ones that they did not study), we show that an ensemble method that we call *EnsInfer* can obtain as good or better results than any single method and improves upon previous ensemble methods ([Marbach et al. 2012; Hill et al. 2016; Saint-Antoine and Singh 2020]). In vision and language applications, some work, such as [Jan and Verma 2020; Shahabadi et al. 2021], uses clustering-based ensemble on large data to create balanced sets which are then sent to distinct learners. In addition to showing the benefits of combining multiple inference methods, our pipeline also provides a practical combination strategy.

Materials and methods

Underlying Network Inference Algorithms

Here we introduce the inference algorithms we used in this ensemble workflow. Our workflow and the open-source code we provide allow the easy incorporation of new inference algorithms.

Experimental Setup: the Data

All level 1 network inference algorithms take gene expression level data as input, there are two main sources for these data: synthetic data generated by simulation software with a given regulatory network or transcriptome-wide RNA sequencing (RNA-seq) data from living organisms. These data can be measured in a temporal manner to constitute time-series data or measured in temporally unrelated discrete states to constitute steady-state data. RNA-seq data can also be classified into two categories: bulk RNA-seq data which is obtained using all cells inside a sample tissue and single-cell RNA-seq data which examines the transcriptome information of a single cell ([Stark et al. 2019]). Details about the gene expression datasets used in our experiments are listed below:

1. Synthetic data from the DREAM3 and DREAM4 *in silico* challenges consists of ten datasets each with 100 genes and varying regulatory network structures. ([Prill et al. 2010; Schaffter et al. 2011]) The gene expression data was generated by GeneNetWeaver, the software which provided data for the DREAM3 and DREAM4 challenges. Simulation settings were kept as the default DREAM4 challenge settings except that we generated five different time intervals between data points: 10 minutes, 20 minutes, 25 minutes, 50 minutes (default value), and 100 minutes. The benefit of using this synthetic data is that the underlying network is precisely known by construction.

2. Bacterial experimental RNA-seq data from B.subtilis (bulk RNA) containing 4218 genes and 239 TFs. The training and testing sets came from a network consisting of 154 TFs and 3144 regulatory edges. ([Arrieta-Ortiz et al. 2015])

3. Plant experimental RNA-seq data (bulk RNA, time-series) from Arabidopsis shoot tissue consisting of 2286 genes and 263 transcription factors (TFs). Both the training and testing sets came from a network consisting of 29 TFs and 4247 regulatory edges. ([Varala et al. 2018])

4. Mouse Embryonic Stem Cell (mESC) experimental single-cell RNA-seq data containing 500 genes and 47 TFs. The training and testing sets came from a functional interaction network consisting of 47 TFs and 3226 regulatory edges. ([Hayashi et al. 2018])

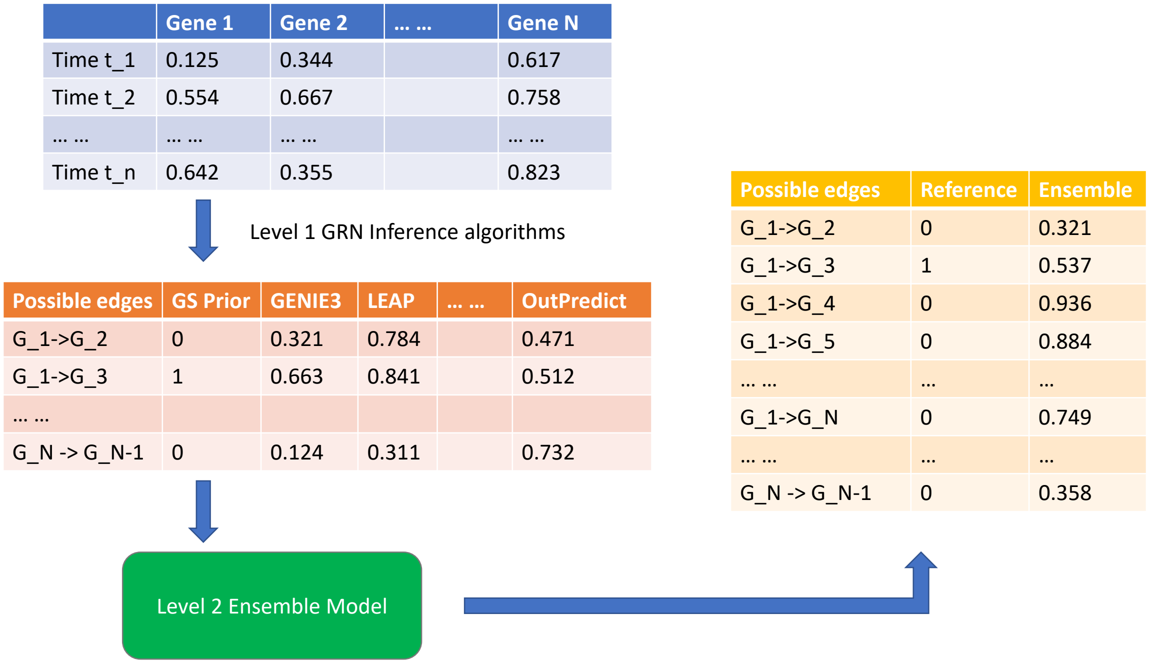
5. Human Embryonic Stem Cell (hESC) experimental single-cell RNA-seq data containing 1115 genes and 130 TFs. The training and testing sets came from a ChiP-Seq network consisting of 130 TFs and 3144 regulatory edges. ([Chu et al. 2016])

In this work, we have focused on either temporal time-series bulk RNA-seq or single cell RNA-seq data for which pseudo-time information is available. One reason is that some of the inference algorithms in the BEELINE framework require temporal information input. The other is the well-known epistemiological reason: steady state data gives simultaneous correlation information, but does not clarify the causal relationship. By contrast, because causation moves forward in time, time series datasets are more useful for causal network inference.

Ensemble Approach

Because one single inference method may not (and, in fact, does not) suit all scenarios, we propose *EnsInfer*, an ensemble approach to the network inference problem: each individual network inference method will work as a first level learning algorithm that gives a set of predictions from the gene expression input. Then we train a second-level ensemble learning algorithm that combines results from those first level learners. As first level inference methods are all different from each other, this forms a heterogeneous stacking ensemble process. ([Wolpert 1992; Aburomman and Reaz 2017]) The end goal is the binary classification task of determining whether or not a potential regulating edge from transcription factor gene to target gene exists.

Thus, base network inference methods such as GENIE3 or Inferelator will work as Level 1 inference methods and individually predict whether some transcription factor TF regulates some target gene g by giving each possible edge a confidence score. The resulting edge predictions of all the level 1 inference methods can be fed into the second level ensemble learner. Previous ensemble approaches include a voting method ([Marbach et al. 2012; Hill et al. 2016], but other approaches have been used for other applications: a random forest classifier, or a Naive Bayesian classifier. The pipeline is shown in Figure 2.1.



	Gene 1	Gene 2	… …	Gene N
Time t_1	0.125	0.344		0.617
Time t_2	0.554	0.667		0.758
… …	… …	… …		… …
Time t_n	0.642	0.355		0.823

Possible edges	GS Prior	GENIE3	LEAP	… …	OutPredict
G_1->G_2	0	0.321	0.784		0.471
G_1->G_3	1	0.663	0.841		0.512
… …					
G_N -> G_N-1	0	0.124	0.311		0.732

Possible edges	Reference	Ensemble
G_1->G_2	0	0.321
G_1->G_3	1	0.537
G_1->G_4	0	0.936
G_1->G_5	0	0.884
… …	…	…
G_1->G_N	0	0.749
… …	…	…
G_N -> G_N-1	0	0.358

Figure 2.1: A diagram showing how *EnsInfer works (i) All level 1 network inference algorithms are executed using time-series expression data. (ii) Every level 1 inference method assigns confidence values to all possible edges in the network. All the outputs are then curated into a tabular form with each algorithm's prediction as a feature column. (iii) The outputs of the level 1 inference methods are then used as input data for the level 2 ensemble model, which makes predictions of regulatory edges.*

Each level 1 inference method infers regulation based on all the given gene expression data. By contrast, the ensemble learner takes a training set consisting of a randomly chosen subset of regulators from gold standard (normally, experimentally verified present/absent) edges and creates a model whose input is the confidence score output of each level 1 inference method and whose output is a prediction about whether each potential edge regulates or not. One thing to note is that, for the sake of consistency across different methods, we use the confidence scores on all regulatory edges of each level 1 inference method not just the highly confident edges. This benefits the level 2 ensemble efforts because all information inferred from level 1 methods is preserved for level 2 models.

The ensemble method uses this model and the outputs of the level 1 inference methods to predict for each transcription factor in the test set, whether a given possible edge leaving that transcription factor corresponds to a true regulatory relation. This process translates well to real world applications, where *EnsInfer* learns from the known regulatory relations within an organism or tissue structure, and makes predictions for untested transcription factors.

We evaluated eight different models to function as level 2 ensemble models using synthetic data. Those models include: voting ([Marbach et al. 2012]), logistic regression, logistic regression with stochastic gradient descent (SGD), Naive Bayes with a Gaussian kernel, support vector machines, k-nearest neighbors, random forest, adaptive boost trees, and XGBoost ([Chen and Guestrin 2016]). All models except XGBoost are provided by the scikit-learn python package ([Pedregosa et al. 2011]). We used a separate DREAM4 dataset with 100 genes to perform hyper-parameter tuning for all level 2 ensemble models. For each of the tunable ensemble models, a discrete set of hyper-parameter combinations spanned by the common selections of core model parameters were cross-validated on this DREAM4 dataset For each method, the best performing hyper-parameter combination was used for the later level 2 comparison experiments. Details about the hyper-parameter grid search and resulting best parameter settings for each model can be found in this Google Drive Link.

We compare the area under the precision-recall curve on the test data of the ensemble learner against that of the level 1 inference methods that have access to the same training data.

Algorithmic Workflow of the Ensemble Approach

All underlying inference algorithms were executed through the BEELINE framework proposed by [Pratapa et al. 2020] to which we added OutPredict and Inferelator which weren't included in the original BEELINE package.

The confidence scores of the underlying algorithms for each potential edge in the regulatory network became inputs to the level 2 ensemble model, as illustrated in Fig 2.1. To compare the performance of different inference methods, we use Area Under the Precision-Recall Curve

(AUPRC) as the primary metric in all experiments. The reason for choosing AUPRC is that experimentalists can choose a high-confidence cutoff to identify the most likely causal transcription factors for a given target gene. A comprehensive summary of the results can be found in Tables 2.1 and 2.2 for experiments on the DREAM *in silico* datasets and Figure 2.2 for the three real-world species.

For the *In silico* DREAM datasets, the underlying gold standard priors that define each regulatory network were divided into a 2:1 training/testing split, so there were twice as many regulators in training as in testing. Because the split was done with respect to the regulators, the training and testing sets share no common transcription factors. We believe splitting based on transcription factors is the correct approach because experimental assays commonly over-express or repress particular transcription factors. The practical goal is that if a species has some TFs with experimentally validated edges, then edges from untested TFs can be inferred.

For each dataset, we first applied 11 base-level inference methods to the training data both to determine a promising single method to apply to the test data and as an input to the construction of the ensemble model. Out of the 12 methods included in BEELINE, SINCERITIES, SINGE, and SCNS either produced no output or exceeded the time limit of one week for one or more of the datasets. We applied those individual level 1 inference methods (not only the most promising ones from the training data) as well as the level 2 non-homogeneous ensemble models to the test set.

To assess ensemble models, we compared them with one another and with the best level 1 inference methods in both training and testing evaluations. As [Marbach et al. 2012] have pointed out for the DREAM challenge, one simple yet (in DREAM at least) effective way to integrate multiple inference results is to rank potential edges according to their average rank given by all inference methods. We will also include this "community" method as a reference point for our ensemble models.

The experiments on the DREAM *in silico* datasets focused on three objectives: (i) for each dataset, how well did the level 1 inference methods that performed best on the training set perform

Table 2.1: Summary of the different gene regulatory networks used in 10 DREAM simulation experiments. The best level 1 inference methods for five different time interval settings (measurements every 10 minutes, every 20 minutes, every 25 minutes, 50 minutes, and 100 minutes) for the time series. Often, the best model on the training set is also the best model on the test set. Qualitatively, when the intervals are shorter, differential equation-style methods are best. When the intervals become larger, random forest methods are often superior.

Number of Edges	10 minutes intervals		20 minutes intervals		25 minutes intervals	
	Best model in training	Best model in testing	Best model in training	Best model in testing	Best model in training	Best model in testing
125	SCRIBE	LEAP	OutPredict	Inferelator	OutPredict	GRISLI
119	OutPredict	OutPredict	Inferelator	Inferelator	Inferelator	OutPredict
166	Inferelator	Inferelator	Inferelator	Inferelator	Inferelator	Inferelator
389	Inferelator	Inferelator	Inferelator	Inferelator	Inferelator	Inferelator
551	Inferelator	Inferelator	Inferelator	Inferelator	Inferelator	Inferelator
176	Inferelator	LEAP	Inferelator	GRNVBEM	Inferelator	Inferelator
249	Inferelator	GRNBoost	Inferelator	GRNBoost	Inferelator	Genie3
195	Inferelator	Inferelator	Inferelator	Inferelator	Inferelator	Inferelator
211	Inferelator	Inferelator	Inferelator	SCRIBE	Inferelator	Inferelator
193	Genie3	Inferelator	Inferelator	Inferelator	Inferelator	Inferelator

a Results for 10, 20, and 25 minutes intervals

Number of Edges	50 minutes intervals		100 minutes intervals	
	Best model in training	Best model in testing	Best model in training	Best model in testing
125	OutPredict	OutPredict	OutPredict	OutPredict
119	OutPredict	OutPredict	OutPredict	Inferelator
166	Inferelator	Inferelator	OutPredict	Inferelator
389	Inferelator	Inferelator	Inferelator	OutPredict
551	Inferelator	Inferelator	Inferelator	Inferelator
176	Inferelator	Inferelator	OutPredict	PIDC
249	OutPredict	OutPredict	Inferelator	Inferelator
195	Inferelator	Inferelator	Inferelator	Inferelator
211	Inferelator	Inferelator	OutPredict	Genie3
193	Inferelator	Inferelator	OutPredict	Inferelator

b Results for 50 and 100 minutes intervals on the test set? (ii) how well did the ensemble learners perform on the test set? (iii) how did the level 1 inference method that performed best on the test set compare to the level 2 ensemble models? Note that the comparison of (iii) is unfair to the ensemble models, because there is no way to know *a priori* which level 1 inference method will perform best on a given test set, so choosing the best one gives an unfair advantage to the level 1 inference methods.

On the experimental datasets from real-world species, similarly, four level 1 methods from BEELINE: GRNVBEM, GRISLI, SINGE, and SCNS were not able to produce proper inference results due to time or memory constraints on the larger datasets (e.g. they did not finish after a week), hence were not included in the ensemble approach. We then applied the best-performing level 2 ensemble models from the DREAM experiments to the available 10 base-level inference methods. Furthermore, we varied the input to the level 2 ensemble models by including or excluding the results from the three most poorly performing level 1 inference methods.

2.2 Results

Base Inference Method Performance

On the DREAM datasets, the performance of the algorithms featured in the BEELINE framework is consistent with the original paper ([Pratapa et al. 2020]). GENIE3, GRNBOOST, and PIDC performed the best among the algorithms the BEELINE authors tested. As it happened, the methods we added to the framework (Inferelator and OutPredict) outperformed those methods in many cases. Nevertheless, no individual level 1 inference method dominated the others, as seen in Table 2.1. We also note that while the best level 1 inference method in training is often the best algorithm in testing, that is not always the case.

Ensemble Performance

The Naive Bayes model we used works on the assumption that the likelihood distribution of edge presence is Gaussian-like with respect to any given input's confidence score. On the DREAM datasets, two of the eleven level 1 inference methods (GRISLI and PIDC) produced outputs that did not have a Gaussian like distribution (reflected as a negative kurtosis). We therefore experimented using all level 1 inference methods as input as well as using only the level 1 inference methods whose output distribution has positive kurtosis as inputs to better accommodate the Naive Bayes model. The combined results are presented in Table 2.2. For most ensemble methods, using all 11 level 1 inference methods versus using 9 does not change the result. However, for Naive Bayes and Logistic Regression with Stochastic Gradient Descent, eliminating those level 1 inference methods that produce non-Gaussian like output helps. In fact, Naive Bayes is overall the winner across all tested models and configurations when the input is limited by the positive kurtosis filter, while logistic regression, random forest, and adaptive boosting also performed favorably compared to the best-performing level 1 inference methods in training as well as compared to the average rank of level 1 inference methods of [Marbach et al. 2012].

Hence, for real-world experimental datasets, the four models: logistic regression, Naive Bayes, random forest, and adaptive boosting were selected as level 2 ensemble models for analysis (see Figure 2.2). Here the likelihood distributions of all results from the level 1 inference methods have a positive kurtosis measure, so all 10 of them were utilized for the level 2 ensemble methods.

Table 2.2: Relative performance of different ensemble methods using all level 1 inference methods' results (i.e., regardless of kurtosis) as ensemble inputs and the same models while only using level 1 inference methods' results with positive kurtosis, marked by plus signs (corresponding to positive kurtosis). Experiments were done across five different DREAM simulation settings for time-series intervals. The performance metric is the ratio of AUPRC score of the ensemble method compared to that of the best performing level 1 inference method in testing. Each cell is the mean value and standard deviation across the ten DREAM datasets with varying regulatory networks. The **bold** number in each column is the best performing value in that time interval setting. Logistic regression, random forest and adaptive boosting models yielded top level inference performance among all ensemble options when there was no filtering based on kurtosis. With kurtosis filtering, the Naive Bayes and logistic regression approaches yield the best overall results while performances from random forest and adaptive boosting are still competitive.

Ensemble Model	Interval between time-series data points				
	10 min	20 min	25 min	50 min	100 min
Logistic Regression	1.11±0.45	0.99±0.21	0.92±0.34	1.33±0.56	**1.25±0.44**
Logistic Regression$^+$	1.13±0.48	1.12±0.43	0.95±0.35	1.31±0.54	1.24±0.46
Logistic Regression with SGD	0.25±0.06	0.4±0.47	0.2±0.11	0.2±0.14	0.26±0.19
Logistic Regression with SGD$^+$	0.71±0.23	0.63±0.19	0.54±0.08	0.55±0.15	0.57±0.21
Naive Bayes	0.83±0.23	0.71±0.17	0.65±0.15	0.67±0.16	0.68±0.26
Naive Bayes$^+$	**1.33±0.58**	1.09±0.26	**1.19±0.5**	1.14±0.48	1.13±0.35
Support Vector Machine	0.21±0.09	0.29±0.14	0.26±0.09	0.25±0.15	0.26±0.22
Support Vector Machine$^+$	0.47±0.2	0.51±0.19	0.43±0.12	0.86±0.6	0.47±0.25
K-Nearest Neighbors	0.59±0.24	0.45±0.22	0.51±0.2	0.39±0.34	0.57±0.41
K-Nearest Neighbors$^+$	0.58±0.23	0.68±0.54	0.71±0.47	0.6±0.46	0.65±0.31
Random Forest	1.12±0.56	0.9±0.21	0.99±0.29	**1.73±1.03**	1.09±0.38
Random Forest$^+$	1.1±0.51	0.97±0.42	1.04±0.37	1.73±1.1	1.08±0.34
Adaptive Boosting	1.12±0.5	**1.13±0.76**	0.73±0.26	1.27±0.65	0.95±0.46
Adaptive Boosting$^+$	1.07±0.43	1.11±0.65	0.82±0.27	1.3±0.7	1.03±0.49
XGBoost	0.65±0.34	0.56±0.16	0.61±0.28	1.35±0.86	0.66±0.24
XGBoost$^+$	0.6±0.25	0.58±0.23	0.78±0.63	1.31±0.86	0.67±0.26
Best level 1 method in training	0.88±0.23	0.88±0.2	0.88±0.24	1±0	0.81±0.22
Best level 1 method evaluated on all samples	0.93 ± 0.26	1.11 ± 0.58	1.08 ± 0.34	1.03 ± 0.49	0.95 ± 0.44
Average rank of level 1 methods	0.95±0.42	0.88±0.52	0.78±0.39	0.69±0.16	0.61±0.22

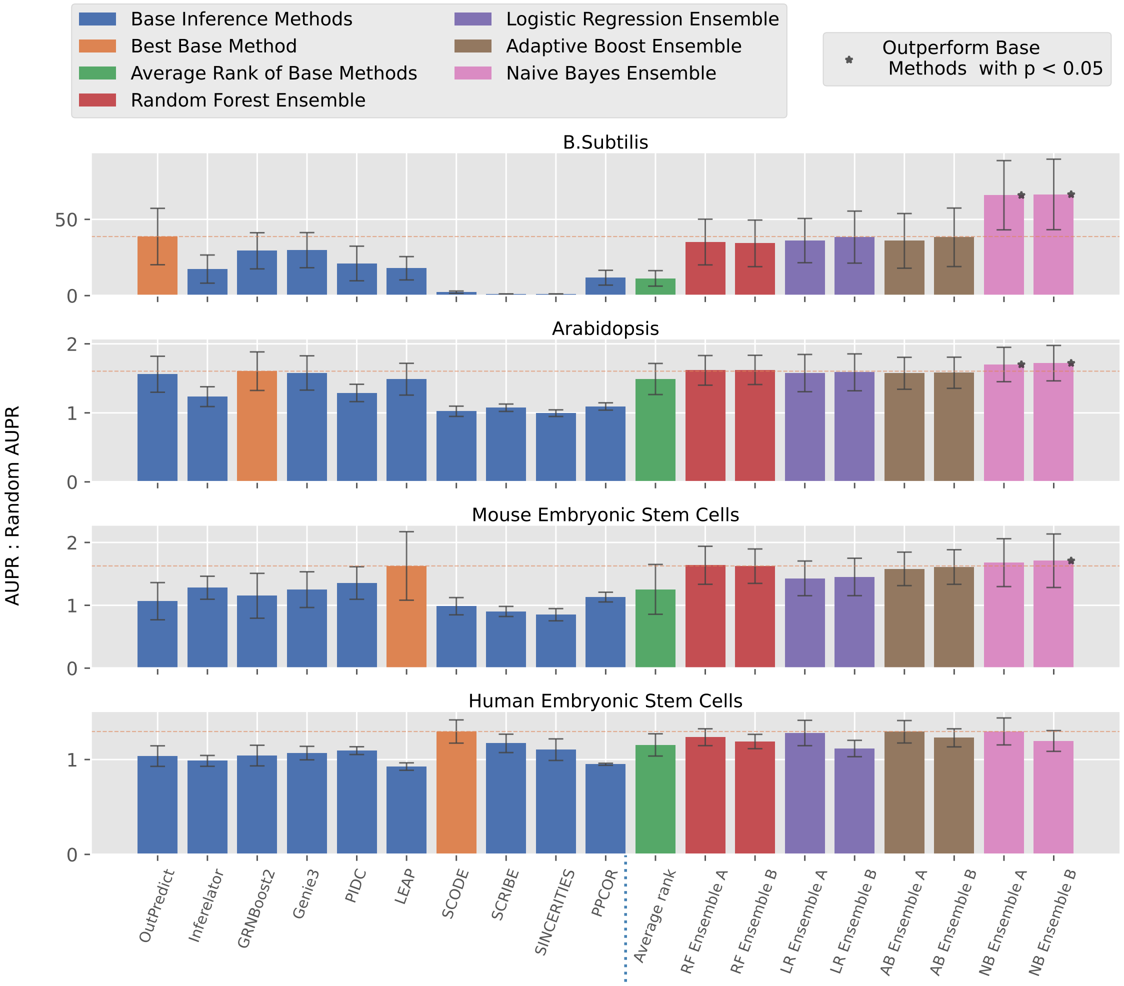


Figure 2.2: The performance of various network inference methods on three different species, from top to bottom: a B. subtilis gene regulatory network using bulk RNA-seq expression data[Arrieta-Ortiz et al. 2015]; an Arabidopsis network using bulk RNA-seq expression data[Varala et al. 2018]; a mouse Embryonic Stem Cell functional interaction network using single-cell RNA-seq data[Hayashi et al. 2018] and a human Embryonic Stem Cell ChIP–seq network using single-cell RNA-seq expression data[Chu et al. 2016]. Inference performance was measured using the ratio of the AUPRC of each inference method divided by that of a random predictor. Gold standard priors from each of the three species were split into a random 2:1 training/testing configuration. Ensemble models along with base inference methods that are able to incorporate prior information were trained using training gold standard priors. Then all inference results were applied using the testing subset of the gold standard data yielding an AUPRC ratio using 20 random training/testing split setups. The mean AUPRC ratio of each method on the test data among these 20 experiments is represented in the bar chart. All four ensemble models were evaluated here with base inference methods, and each of them was trained with the three worst performing base inference methods in the training set (see the A series histogram) or without the three worst performing base inference methods (B series histogram). Asterisks indicate a statistically significant (p-value below 0.05 in non-parametric paired tests) improvement compared to the best level 1 inference method (and compared to the average ranking approach). Overall, Naive Bayes performs best, but in some cases Adaptive Boosting and Random Forests do almost as well.

Figure 2.2 shows that

- The Naive Bayes approach on inputs having positive kurtosis outperforms the other three ensemble methods, so our system *EnsInfer* uses Naive Bayes as the default option.
- Including results from weak learners has a marginal impact (sometimes positive and sometimes negative) on the final ensemble performance. For the sake of simplicity, therefore, *EnsInfer* includes inputs from all available inference methods having positive kurtosis, even the weak ones.

Figure 2.3 shows that the Naive Bayes ensemble approach significantly (p-value < 0.05) outperformed the best level 1 method on B.subtilis and Arabidopsis. The ensemble method with all level 1 methods still had an advantage in mESC data although the performance gain was less statistically significant with a p-value of 0.133. To calculate the p-value, we conservatively chose a non-parametric paired resampling approach ([Shasha and Wilson 2010]) because we did not want to assume any particular distribution on the data. We used a paired test because we measured the AUPRC gain for each training/test split. (That is, the set of training/testing splits were established randomly and initially. Then, for the numerical experiments, each method used that set.) In the hESC case, the Naive Bayes ensemble method achieved approximately the same level of performance as the best level 1 method on the test set. As noted above, the best level 1 inference method for the test set cannot be known *a priori* (and not even looking at each method's performance in training), so using an ensemble method gives high performance without having to know which level 1 inference method is best. The Naive Bayes approach also consistently outperformed the average voting ensemble approach ([Marbach et al. 2012]).

EnsInfer: Compared to running a single inference method, the ensemble approach requires an amount of computation resources equal to the sum of the time to run all base inference algorithms, plus the ensemble effort itself. However, all base inference methods can be executed in parallel, so the wall clock time of executing level 1 inference process is just the time of the slowest method

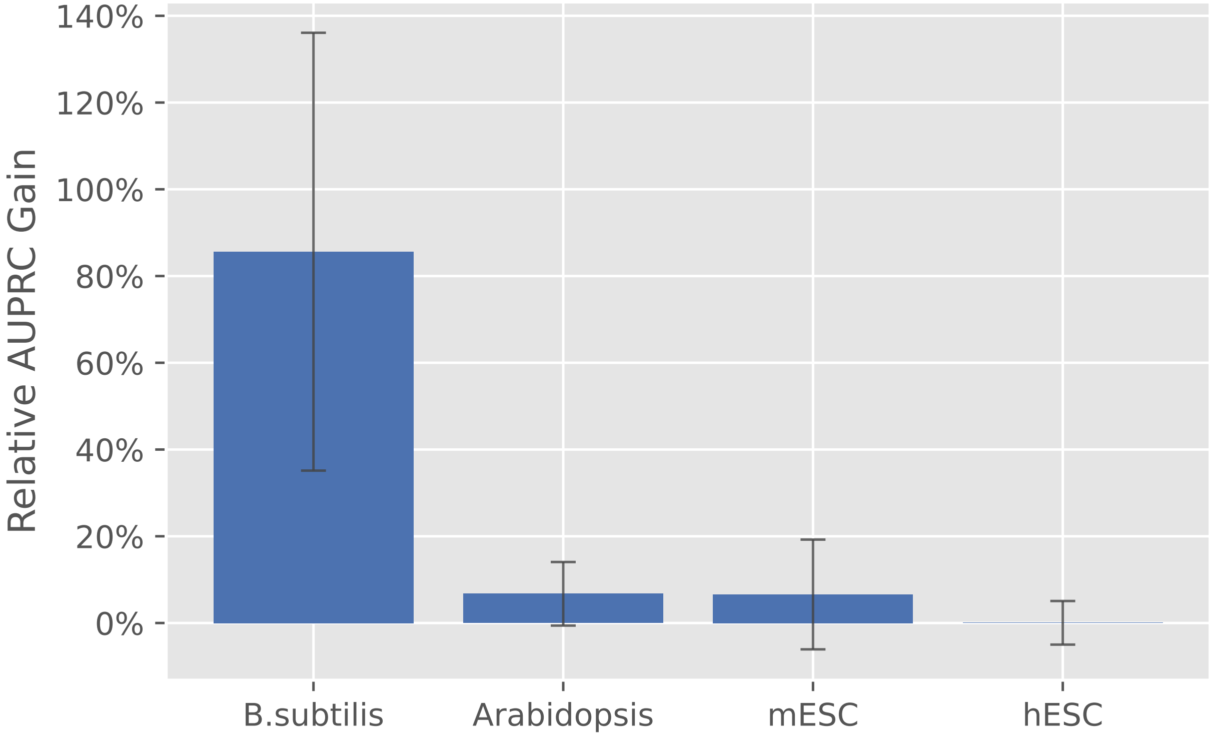


Figure 2.3: The AUPRC improvement of the Naive Bayes ensemble model (restricting inputs to those with positive kurtosis, but including weak learners) compared to the single best base inference method. In the B.subtilis and Arabidopsis datasets, the improvement had a p-value < 0.05 using a non-parametric paired test.[Shasha and Wilson 2010] The test should be paired, because the same set of training and testing splits were used for every method. In the human dataset, the ensemble method was about equal to the best base inference method. As noted in the text, the best base method cannot be known *a priori*, so these comparisons understate the advantage of the ensemble method.

which often is also single-threaded. The level 2 ensemble effort itself is less than 1/10 the time of the slowest base method as shown in Table 2.3. We can therefore conclude that *EnsInfer*'s wall clock time is close to that of the slowest base inference method.

2.3 Discussion

Consistently with [Pratapa et al. 2020], we find that no one inference method is best for all datasets tested in our study. However, a Naive Bayes level 2 ensemble model built from level 1 inference methods having positive kurtosis holds great promise as a general ensemble network

Table 2.3: Execution time of all the inference methods used in the study, across different datasets. All computation programs were executed in a Ubuntu 20.04 environment with a single AMD Ryzen™ 9 5900X processor and 64 GB of system memory.

Inference Method	Execution time, measured in seconds			
	mESC	hESC	Arabidopsis	Bsubtilis
OutPredict	95.72	134.46	182.03	793.85
Inferelater	21.48	44.05	58.31	172.74
GRNBoost 2	22.46	38.07	106.38	312.51
Genie3	148.53	387.93	1174.46	3652
PIDC	40.89	166.18	1376.25	3764
LEAP	8.42	124.54	35.78	705.23
SCODE	1883.54	7705	8288	11252
SCRIBE	1672.2	25206	29090	46829
SINCERITIES	38.47	275.71	2749.83	38641
PPCOR	2.07	10.45	47.79	280.00
Level 2 Random Foreset Ensemble	2.74	11.87	18.10	128.57
Level 2 Logistic Regression Ensemble	0.24	1.28	9.09	9.93
Level 2 Adaboost Ensemble	2.68	19.06	33.28	156.47
Level 2 Naive Bayes Ensemble	0.05	0.35	8.46	4.32

inference approach and is thus the basis of *EnsInfer*. Naive Bayes may work better than more sophisticated Bayesian methods because at the core of the Bayesian method, we need to estimate the likelihood distribution of $p(x|e)$ where x is the score given by a level 1 inference method and e is the existence of an edge. Since this generative process varies dramatically across different datasets and inference methods, the Gaussian assumption used by Naive Bayes is as good as any and keeps the model simple.

Please note, however, that there are cases when Naive Bayes does not improve on the best single inference method. This happens primarily when the results are a little better than random. For example, the inferred regulatory networks from single-cell human embryonic stem cell data from [Pratapa et al. 2020] were barely better than random using any base method in BEELINE. The ensemble does not improve that.

Naive Bayes works particularly well in a sparse data environment, which is often the case when experimental data is hard to come by. For example, there are only 29 experimentally validated transcription factors for Arabidopsis and 154 for B. Subtilis. Another point in favor of Naive Bayes is that the size of the feature space (the number of outputs of the level 1 inference methods) is small. If the training dataset and feature space were larger, Random Forest-based approaches might do better. Our current investigation used roughly a dozen level 1 inference methods. Other promising new inference ones could be added such as BiXGBoost and DeepSEM [Zheng et al. 2019; Shu et al. 2021]. A level 2 ensemble method might potentially require a feature selection step if many more inference algorithms were included.

Conclusion

The main overall benefit of the ensemble method *EnsInfer* is its robust and flexible nature. Instead of picking a network inference method and hoping that it will perform well on a dataset, *EnsInfer* uses a combination of state-of-the-art inference approaches and combines them using a simple Naive Bayes ensemble model. Because the ensemble approach essentially turns all the predictions from different inference algorithms into priors about each edge in the network, *EnsInfer* easily allows the integration of diverse kinds of data (e.g. bulk RNA-seq, single-cell RNA-seq) as well as new inference methods.

3 | Bipartite Networks

3.1 Background and Motivation

A frequent goal of expression-based causality research is to construct a directed graph of genes having some inductive and some repressive edges. One of the popular approaches to solving the gene regulatory inference problem is to build some kind of regression models to fit the gene expression data and make regulatory relation inferences based on the model parameters. Some of these regression-based methods use linear regression like TIGRESS [Haury et al. 2012], SCODE [Matsumoto et al. 2017] and Inferelator [Skok Gibbs et al. 2022]. While others like GENIE3 [Huynh-Thu et al. 2010], Bixgboost [Zheng et al. 2019], OutPredict [Cirrone et al. 2020] and SCENIC [Van de Sande et al. 2020] chose non-linear regression models for the same goal. The main metric of these methods is conformance to some gold standard network. But let's consider the actionable result of such research: to influence the behavior of an organism to make it more useful (e.g. more drought-tolerant crop or one with higher nutrient yield).

This paper asks the question "Are networks a good representation for actionable insights?" Because the edges are simple edges between pairs of regulatory and target genes, the network representation does not suggest any kind of synergy between the putative causal regulatory genes, e.g., transcription factors. So the natural model choice for a given target gene g given the network is a linear model on the genes pointing to g.

Any causality model should be able to make reasonably accurate predictions. In Newtonian mechanics, for example, a model involving mass and gravity will be able to predict the speed curve of a ball on an inclined plane. Predictive accuracy is not **sufficient** to establish causality. Some mysterious force might cause the ball to move in that way, but a causal model should be predictive. We consider predictive accuracy to be a **necessary** condition of a causal model.

We now consider several approaches to prediction.

1. Starting with all transcription factors (TFs) as possible causal features, try both a non-linear random forest-style model M_{nonlin} and a linear model M_{lin}.

2. Based on the random forest model above, iteratively search for a minimal set of transcription factors (TFs) that could produce similar regression results in a model $M_{minimal}$

3. On the edges from a gold standard network for some target gene g, use a non-linear model M_{nonlin} or a linear model M_{lin}.

4. Form a random forest model that uses the same number of TFs for each target g as known from the gold standard network. Choose those TFs according to their feature importance starting from M_{nonlin} in approach 1 above.

As we will see: (i) the non-linear models work better than the linear models, and (ii) starting with all transcription factors and then shrinking that set based on model accuracy is better than using the gold standard network.

Our models are all based on time series in which we predict the mRNA expression level of the target gene based on the state of regulatory genes at the previous time point. This accords with the biological intuition that the state of causal regulatory genes takes at least minutes to affect their targets. One implication of this approach to model building is that if a transcription factor T and a target gene g are correlated (e.g. they rise in the same time points and fall in the same time points), T will **not** be identified as causal. On the other hand, if T rising (respectively, falling) at one time point were associated with g rising (respectively, falling) at the next time point, then causality might be hypothesized.

3.1.1 Contributions

Our novel contribution is a framework, algorithms, and software for encoding possible causality in transcriptional settings into a bipartite directed graph. Our framework consists of the following workflow:

1. Choose a machine learning method M that predicts the behavior of a target gene g starting with all possible causal regulatory elements (transcription factors for genomic networks) as candidates. M may be statistical, a neural network, a forest, or a linear model. We do not advocate any particular model, though non-linear models generally have lower errors than linear ones.

2. Reduce the set of possible causal regulatory elements for g to a smaller set S_1 that gives statistically equal (based on p-value) accuracy still based on M.

3. Inspired by an observation of the statistician Efron [Efron 2020], find a set D of mutually disjoint sets $S_1, S_2 ... S_n$ that all give statistically the same accuracy as S_1 in predicting the expression of some target gene g, possibly by training a new model for each S_i. D may contain S_1 alone or may contain many sets.

4. Provide a visual representation of these mutually disjoint subsets of regulatory elements for each given target gene g. The visualization consists of a bipartite graph in which each transcription factor of the disjoint subsets ($S_1, S_2 ... S_n$) of transcription factors feeds a model node whose output is the target gene g.

3.2 Materials and methods

3.2.1 Expression Prediction Setup

For all the experiments carried out in this study, we focus on time series RNA-seq data, because gene regulation through transcription factors is a temporal causal process. Following this logic, we build regression models that predict each target gene's expression based on the TF expression levels from a previous time point in the time series. Formally, suppose we are given a time series RNA-seq data consisting of sequencing data from time points $t_0, t_1, t_2, ..., t_i, ..., t_n$. We use RNA-seq data at time t_i to predict the expression of a target gene at t_{i+1} $(i \geq 0, i < n)$.

In order to split the whole time series into training/testing sets for validating the prediction quality of different methods. We chose to always reserve the tail end of the time series for testing while using the preceding part in training. More specifically, if $n < 5$, then only t_n will be used as a test sample with t_{n-1} being the input. If $5 \leq n < 10$, t_n and t_{n-1} are reserved for testing with t_{n-1}, t_{n-2} as model input. If $n \geq 10$, then the final three in the series constitute the testing set.

3.2.2 RNA Sequencing Data Used

Bulk time-series RNA-seq data from four different species with varying experimental setups were used, totaling more than 100 data points for each species. Here are the experimental sources for each species. For the training/testing setups below, we train on a prefix of the time points and test on the remaining time points.

1. Saccharomyces cerevisiae (yeast): data from GSE145936 [Feder et al. 2021], GSE153609 [Mitra et al. 2021; Tran et al. 2021], GSE168699 [Li et al. 2021] and GSE226769[Harris and Ünal 2023] were aggregated into a gene expression dataset with 144 training samples and 58 testing samples.

2. Bacillus subtilis (strain 168) (B.subtilis): data from GSE108659 [Krawczyk et al. 2015], GSE128875 [Pisithkul et al. 2019] and GSE224332 were aggregated into a gene expression dataset with 84 training samples and 18 testing samples

3. Arabidopsis thaliana (Arabidopsis): data from GSE97500 [Varala et al. 2018; Heerah et al. 2021] was used with 72 training samples and 24 testing samples.

4. Mus musculus (mouse): data from GSE115553 [Graham et al. 2018], GSE151173 [Greenwell et al. 2020] and GSE171975 [Aviram et al. 2021] aggregated into a gene expression dataset with 208 training samples and 121 testing samples.

5. Homo sapiens (human): data from GSE221103 and GSE221173 [Cazarin et al. 2023] were aggregated into a gene expression dataset consisting of 109 training samples and 40 testing samples.

Thus, our study derives from four well-studied living organisms ranging from bacteria to humans. We sought data sets having time series RNA-seq data with relatively tight timing intervals (no greater than 4 hours in all cases) and suitably long series ($\geq$ 4 time points) to form training/testing splits. We collected as much public bulk RNA-seq data about the four species as possible given the above constraints. Because the data came from widely different species (from bacteria to humans), we expect that our qualitative conclusions will be generalizable. For each species, we obtained a Gold Standard (GS) regulatory network from the sources listed in Table 3.1.

3.2.3 Metrics

Because RNA-seq counts are strongly dependent on the amount of cellular material that is read, relative expression is a better metric to determine induction or repression than absolute expression. For that reason, we measure expression based on the z-score of the normalized RNA-seq counts in

Table 3.1: Information on the Gold Standard (GS) networks used in this study. The number of target genes and transcription factors are for genes that are both present in the regulatory network and the RNA-seq data, for each species respectively.

Species	**GS Network Source**	**Number of Target Genes**	**Number of TFs**	**Number of Regulations**
Yeast	Yeastract[Teixeira et al. 2023]	4794	213	162100
B.subtilis	SubtiWiki[Pedreira et al. 2022]	1878	146	3973
Arabidopsis	ConnecTF[Brooks et al. 2021]	18855	57	141445
Mouse	RegNetwork[Liu et al. 2015]	8211	780	40331
Human	RegNetwork[Liu et al. 2015]	17533	1351	132259

the form of TPM (Transcripts Per Kilobase Million):

$$z = \frac{TPM - \mu}{\sigma} \tag{3.1}$$

To compare the performance of each method, we measure how accurate the regression results were by checking the error of the prediction on the test set for each of the target genes. More specifically, the root mean square error (RMSE) of the model prediction in the test set for each target gene's expression was compared across different regression models. Because every regression model was trained/fitted to make predictions on the same set of time series expression samples for each target gene in question, the performance metrics can be compared based on a paired test. For this purpose, we use a non-parametric paired test [Katari et al. 2021].

3.2.4 Methods Compared

For the purpose of predicting each target gene's expression on a future unseen time point, we fitted four different types of regression models, as described above:

1. A random forest (RF) model that takes the expression of all transcription factors (TFs) as input.

2. A ridge regression (linear regression with L2 regularization) model that takes the expression

of all transcription factors (TFs) as input.

3. A random forest model that takes only the expression of TFs known from the gold standard (GS) network for each particular target gene as input.

4. A ridge regression model that takes only the expression of TFs known from the gold standard (GS) network for each particular target gene as input.

Next, we test how good the transcription factors from the gold standard network are compared to the same number of transcription factors derived from a non-linear model. For each target gene g, let k_g be the number of transcription factors in the gold standard network that point to g. In addition to the tests above, we compare a random forest on those gold standard transcription factors against a random forest for g based on the top k_g transcription factors found using method 1 above. The idea is to test the usefulness of gold standard edges for prediction. One may argue that gold standard edges are inferred using different methods – usually by modifying single regulatory genes and seeing their effect – and therefore should not necessarily be useful for prediction, but could still be useful if modifying a single gene is all that is possible for practical reasons. We do not contest their utility for such purposes. We do however want to evaluate their predictive power in a synergistic setting (i.e. when potentially several regulatory genes can be simultaneously modified).

Finally, using the method of section 3.3, we construct a minimal random forest model for each target gene g on the training set and view its result on the test set. We chose random forest because decision tree-based regression models have proven to be among the best methods in gene regulatory tasks.[Huynh-Thu et al. 2010; Moerman et al. 2019; Zheng et al. 2019] We didn't expand our model selection, because the main focus of our work is not to find the best-fitted machine learning model for the task but rather to demonstrate a novel approach to the representation of potential causality in gene regulation.

3.3 Algorithms to Construct Bipartite Causality Graphs

Having chosen prediction as the metric for causality, we now turn to the other three goals of our proposed framework:

1. Find minimal sets of predictive TFs

2. Find disjoint minimal sets that have p-value-indistinguishable predictive accuracy.

3. Create a bipartite visual representation of causality.

3.3.1 Feature Importance of Random Forest

One fundamental part for our proposed bipartite representation of causality is the feature importance metric used in random forest models. It refers to the measures that quantify how much each feature contributes to the predictive performance of the model. The most common one being the built-in Gini importance that measures the total decrease in Gini impurity when a feature is used for splitting across all trees in the forest. First we start with the definition of the Gini index at each tree node v:

$$Gini(v) = \sum_{c=1}^{C} p_c^v (1 - p_c^v) \tag{3.2}$$

Where p_c^v is the proportion of samples of class c at node v. A lower Gini impurity indicates a purer node (more samples of a single class). If node v uses feature X_i for the split of its two children nodes v_L and v_R, the Gini impurity gain of feature X_i at this node is then given by:

$$Gain(X_i, v) = Gini(X_i, v) - \rho_L Gini(X_i, v_L) - \rho_R Gini(X_i, v_R) \tag{3.3}$$

Where ρ_L and ρ_R are the proportions of samples going to each child.

Then for each feature, its Gini importance is simply calculated by summing the Gini gains for all nodes where that feature is used to split, across all trees in the forest. Features that are used for splits more often and/or result in larger Gini gains will have higher importance scores. Often, these raw importance scores are normalized so that they sum to 1 or 100%, making them easier to interpret as relative importances.

Gini importance is computationally efficient as it's calculated during the training process, thus providing a measure of feature importance immediately after the random forest model is fitted to the training data. As the importance metric ranks all the features based on their individual influence on the model, it forms the basis of our iterative feature selection process that leads to the minimal sets of predictive TFs. It can also capture non-linear relationships and interactions between features which is the main advantage of a bipartite representation over traditional network representation.

3.3.2 Minimal Sets of Predictive Transcription Factors

In [Efron 2020], Efron notes that disjoint sets of causal factors often enjoy similar predictive accuracy. Inspired by this observation, we propose the following strategy. For a given target gene, start by fitting a random forest predictor that takes the expression levels of all known TFs to predict the expression of the target. Then the number of TFs are iteratively cut in half based on their feature importance in the fitted model until a further reduction results in a statistically significant (p-value < 0.05) worse performing random forest. We refer to the final remaining set of TFs as “Minimal TF Set per Target". The pseudo-code for this feature selection process is shown in Algorithm 1.

The histograms in Figure 3.1 show the distribution of the size of minimal TF sets yielded for each target gene for the four species we investigated. From these size distributions, we can see that most of the minimal sets consist of a rather small number of TFs. When compared with the distribution of gold standard network coverage for each target gene in Table 3.2, we see that an accurate regression model constructed this way usually has fewer input transcription factors compared to the gold standard networks.

Algorithm 1 Minimal Transcription Factor (TF) set per Target: For each target gene G, initial set of TFs, and Root Mean Squared Error E, repeatedly reduce the set of necessary transcription factors by half until the error grows significantly with respect to E. Minimal set of TFs for a given target gene G will be a call to this function MinimalSet(G, $all\ TFs$, 0)

```
function MinimalSet($G, TFs, E$)
    $F \leftarrow TFs$
    $M_{all} \leftarrow$ initialized regression model
    Fit $M_{all}$ with $F$ to predict $G$
    if $E > 0$ then
        $E_{baseline} \leftarrow E$
    else                          ▷ $E == 0$ implies that no error value has been calculated yet
        $E_{baseline} \leftarrow$ training Error of $M_{all}$
    $F_{half} \leftarrow$ top half most influential TFs used in $M_{all}$
    $flag \leftarrow True$
    while $flag == True$ do
        $M_{current} \leftarrow$ initialized regression model
        Fit $M_{current}$ with $F_{half}$ to predict target gene $G$
        $E_{current} \leftarrow$ training Error of $M_{current}$
        if not $E_{current} > E_{baseline}$ with statistical significance then
            $F \leftarrow F_{half}$
            $F_{half} \leftarrow$ top half most influential TFs used in $M_{current}$
        else
            $flag \leftarrow False$
    return $F$
```

3.3.3 Finding Disjoint Sets of Predictive Transaction Factors

After finding a minimal set of predictive transcription factors, our algorithm performs a new round of TF searches to discover disjoint sets of roughly equally predictive TFs. Algorithm 2 describes the process for finding all such disjoint sets of a given target gene. Similar to the minimal set search algorithm, we based our iterative search on the random forest regression that takes all available TFs U as input. Rather than stopping after a minimal set $S1$ is found, we test if using all remaining TFs ($U - S1$) could also produce a regression prediction as good as the baseline. If that

Table 3.2: Gold standard network edge coverage per target gene for different RNA-seq species compared with the size of minimal transcription factor set sizes per target gene, derived using random forest regression. Distributions are presented as the median followed by the interquartile range (IQR) which is the range between the 25th and 75th percentile of the data. The one exception is yeast, where the gold standard edges are much more numerous. As shown in the tables above, the minimal sets generally have better predictive power than the gold standard sets and roughly the same number of edges per target.

Median [IQR]	**Yeast**	**B.subtilis**	**Arabidopsis**	**Mouse**	**Human**
Size of TF set per Target in the Gold Standard Network	51 [41, 63]	1 [1, 2]	13 [6, 19]	4 [2, 7]	8 [4, 14]
Size of Minimal TF Set Per Target Using RF Regression	3 [3, 3]	2 [2, 9]	3 [3, 3]	5 [2, 10]	3 [3, 3]
Number of Target Genes	385	733	1373	310	698

is the case, we carry on a similar feature reduction process that ends with a new "minimal set" $S2$ from $U - S1$. This process then repeats with $U - (S1 \cup S2)$ and continues until the baseline performance cannot be beaten or there are no TFs left. For each target gene g, we define the discovered collection of minimal sets discovered this way as the *minimal disjoint sets of predictive transcription factors for g* or *MinDisjoints(g)* for short.

We then surveyed the distribution of how many MinDisjoints are found for each target gene across the four species. The histograms in Figure 3.2 show that most of the target genes have a small number of disjoint sets of TFs associated with them, while some target genes have a large number of MinDisjoints. Our analysis does not yield biological mechanisms for predictability/causality, so we have no mechanistic explanation for how multiple disjoint sets of transcription factors might control the same target gene. However, the result is not wholly unexpected given the well-known redundancy in biological systems.

3.3.4 Bipartite Network Representation

Networks have a pleasing visual representation, especially when focussing on one or a few target genes. What we have shown though is that the network itself is a poor basis for prediction. Now that we have constructed multiple disjoint sets of predictive TFs for each target gene g, we

Algorithm 2 Disjoint sets of Transcription Factors (TFs): Find minimal sets of disjoint TFs (MinDisjoints) where each minimal set has the same error as using all TFs.

```
$D \longleftarrow$ empty list
$F \longleftarrow$ All TFs
$M_{all} \longleftarrow$ initialized regression model
Fit $M_{all}$ with $F$ to predict target gene $G$
$E_{all} \longleftarrow$ training Error of $M_{all}$
$F_m \longleftarrow \textsc{MinimalSet}(G, F, E_{all})$
Add $F_m$ to $D$
$F_r \longleftarrow F \setminus F_m$
$flag \longleftarrow True$
while $F_r \neq$ and $flag == True$ do
    $M_r \longleftarrow$ initialized regression model
    Fit $M_r$ with $F_r$ to predict target gene $G$
    $E_r \longleftarrow$ training Error of $M_r$
    if $E_r > E_{all}$, with statistical significance then
        $flag \longleftarrow False$
        break
    $F \longleftarrow \textsc{MinimalSet}(G, F_r, E_{all})$
    Add $F$ to $D$
    $F_r \longleftarrow F_r \setminus F$
For a given target gene $G$, $D$ will be the set of disjoint sets of TFs $G$.
```

propose a bipartite representation for them. The bipartite representation for each target gene g consists of a model node m corresponding to each disjoint set d_m from D(g). The TFs from d_m in turn point to m.

Suppose that TFs A, B, and C through model M(A,B,C) give good predictions regarding target gene g. Suppose further that TFs D, E, F, and H give roughly equally good predictions on g. The classic gene regulatory approach would be a graph with arrows from A, B, C, D, E, F, and H all pointing to g. The Bipartite approach would suggest instead to show a bipartite graph that would have A, B, and C point to a model node which in turn points to g, and have D, E, F, and H point to a different model node which also points to g.

To demonstrate this new representation, we picked one example for each species we studied, as shown in Figure 3.3. Here we specifically highlighted one interesting scenario: a set of TFs were found to form one of the disjoint sets for more than one target gene. Such a relationship between two genes would not have been found in a simple network representation. The bipartite representation reveals group effects that would not otherwise be evident.

3.4 Results

3.4.1 Comparison of Approaches

Figure 3.4 shows the accuracy of the six different modeling approaches listed in Section 3.2.4. Basically feeding expression information from all the TFs into a random forest (“RF with all TFs") yielded the best outcome. Relying solely on known Gold Standard edges (“RF with GS TFs") usually performed poorly, even compared to using the same number of TFs for each target gene from the random forest model (“RF with top TFs").

We note that linear models on all TFs are competitive and sometimes better than Random Forests on minimal TF sets for B subtilis and mouse. Still, over all, given the same input information, Random Forests perform better than linear models, which is the main point of that comparison.

Table 3.3, 3.4, 3.5, 3.6 and 3.7 lists the detailed pairwise non-parametric results comparing the performance of all possible pairs of models. The tables show that using all TFs in the regression yields the highest prediction accuracy. Finding a minimal set of the most important TFs yields almost the same accuracy as using all TFs.

Table 3.3: S. cerevisieae (Yeast): Paired non-parametric results for the performance comparison on the test set between the model in blue and orange measured using root mean square error (RMSE) on target gene expressions in the test datasets. Entry (i, j) shows the 95% confidence interval as the difference of the ith blue modeling method minus the jth orange modeling method. When the difference in the blue method i with the orange method j has a p-value below 0.05 based on a non-parametric paired test, the (i, j)th confidence interval will be colored accordingly. A negative number means the method in blue has a lower error so is better, and vice versa. Otherwise the (i, j)th entry will be black.

Glossary: (i) TF = transcription factor, (ii) GS = gold standard edges, (iii) RF = Random Forest, (iv) linear = ridge regression, (v) RF with top TFs = for each target gene g, use the same number of TFs from the random forest model as there were gold standard edges for g, and (vi) minimal TF = minimal set of most important TFs that gives p-value-indistinguishable results (on training set) using all TFs.

	RF with all TFs	Linear with all TFs	RF with GS TFs	Linear with GS TFs	RF with top TFs	RF with minimal TF set
Mean RMSE	0.877	1.204	0.920	1.091	0.859	0.965
RF with all TFs	-	(-0.397, -0.257)	(-0.055, -0.031)	(-0.263, -0.166)	(0.008, 0.027)	(-0.119, -0.058)
Linear with all TFs	(0.257, 0.397)	-	(0.214, 0.354)	(0.047, 0.178)	(0.275, 0.414)	(0.169, 0.309)
RF with GS TFs	(0.031, 0.055)	(-0.354, -0.214)	-	(-0.218, -0.124)	(0.046, 0.076)	(-0.075, -0.015)
Linear with GS TFs	(0.166, 0.263)	(-0.178, -0.047)	(0.124, 0.218)	-	(0.185, 0.279)	(0.081, 0.172)
RF with top TFs	(-0.027, -0.008)	(-0.414, -0.275)	(-0.076, -0.046)	(-0.279, -0.185)	-	(-0.133, -0.079)
RF with minimal TF set	(0.058, 0.119)	(-0.309, -0.169)	(0.015, 0.075)	(-0.172, -0.081)	(0.079, 0.133)	-

A question to ask is what biological meaning the disjoint sets could have for a given target gene g. Our computational analysis does not give a biological meaning other than predictive ability. Experimentalists might take various disjoint sets of transcription factors and manipulate them to achieve some desired effect on a target gene. The choice of such sets may depend on the

Table 3.4: B.subtilis: Paired non-parametric results for the performance comparison between the model in blue and orange measured using root mean square error (RMSE) on target gene expressions in the test datasets. Entry (i, j) shows the 95% confidence interval as the difference of the ith blue modeling method minus the jth orange modeling method. When the difference in the blue method i with the orange method j has a p-value below 0.05 based on a non-parametric paired test, the (i, j)th confidence interval will be colored accordingly. A negative number means the method in blue has a lower error so is better, and vice versa. Otherwise, the (i, j)th entry will be black. Please see the caption of Table 3.3 for a glossary of terms.

	RF with all TFs	Linear with all TFs	RF with GS TFs	Linear with GS TFs	RF with top TFs	RF with minimal TF set
Mean RMSE	0.454	0.469	0.599	0.675	0.556	0.489
RF with all TFs	-	(-0.035, 0.003)	(-0.168, -0.122)	(-0.247, -0.196)	(-0.124, -0.080)	(-0.053, -0.017)
Linear with all TFs	(-0.003, 0.035)	-	(-0.161, -0.098)	(-0.243, -0.168)	(-0.113, -0.060)	(-0.037, -0.001)
RF with GS TFs	(0.122, 0.168)	(0.098, 0.161)	-	(-0.099, -0.054)	(0.015, 0.071)	(0.080, 0.141)
Linear with GS TFs	(0.196, 0.247)	(0.168, 0.243)	(0.054, 0.099)	-	(0.086, 0.153)	(0.153, 0.221)
RF with top TFs	(0.080, 0.124)	(0.060, 0.113)	(-0.071, -0.015)	(-0.153, -0.086)	-	(0.044, 0.091)
RF with minimal TF set	(0.017, 0.053)	(0.001, 0.037)	(-0.141, -0.080)	(-0.221, -0.153)	(-0.091, -0.044)	-

Table 3.5: Arabidopsis: Paired non-parametric results for the performance comparison between the model in blue and orange measured using root mean square error (RMSE) on target gene expressions in the test datasets. Entry (i, j) shows the 95% confidence interval as the difference of the ith blue modeling method minus the jth orange modeling method. When the difference in the blue method i with the orange method j has a p-value below 0.05 based on a non-parametric paired test, the (i, j)th confidence interval will be colored accordingly. A negative number means the method in blue has a lower error so is better, and vice versa. Otherwise, the (i, j)th entry will be black. Please see the caption of Table 3.3 for a glossary of terms.

	RF with all TFs	Linear with all TFs	RF with GS TFs	Linear with GS TFs	RF with top TFs	RF with minimal TF set
Mean RMSE	0.851	1.157	0.964	0.978	0.874	0.919
RF with all TFs	-	(-0.345, -0.268)	(-0.135, -0.092)	(-0.148, -0.108)	(-0.032, -0.014)	(-0.080, -0.057)
Linear with all TFs	(0.268, 0.345)	-	(0.159, 0.227)	(0.142, 0.215)	(0.245, 0.322)	(0.203, 0.274)
RF with GS TFs	(0.092, 0.135)	(-0.227, -0.159)	-	(-0.028, -0.001)	(0.069, 0.112)	(0.023, 0.067)
Linear with GS TFs	(0.108, 0.148)	(-0.215, -0.142)	(0.001, 0.028)	-	(0.086, 0.124)	(0.039, 0.081)
RF with top TFs	(0.014, 0.032)	(-0.322, -0.245)	(-0.112, -0.069)	(-0.124, -0.086)	-	(-0.057, -0.033)
RF with minimal TF set	(0.057, 0.080)	(-0.274, -0.203)	(-0.067, -0.023)	(-0.081, -0.039)	(0.033, 0.057)	-

Table 3.6: Mouse: Paired non-parametric results for the performance comparison between the model in blue and orange measured using root mean square error (RMSE) on target gene expressions in the test datasets. Entry (i, j) shows the 95% confidence interval as the difference of the ith blue modeling method minus the jth orange modeling method. When the difference in the blue method i with the orange method j has a p-value below 0.05 based on a non-parametric paired test, the (i, j)th confidence interval will be colored accordingly. A negative number means the method in blue has a lower error so is better, and vice versa. Otherwise the (i, j)th entry will be black. Please see the caption of Table 3.3 for a glossary of terms.

	RF with all TFs	Linear with all TFs	RF with GS TFs	Linear with GS TFs	RF with top TFs	RF with minimal TF set
Mean RMSE	1.194	1.305	1.480	1.686	1.331	1.293
RF with all TFs	-	(-0.217, -0.006)	(-0.437, -0.136)	(-0.669, -0.316)	(-0.200, -0.074)	(-0.169, -0.029)
Linear with all TFs	(0.006, 0.217)	-	(-0.261, -0.088)	(-0.503, -0.259)	(-0.110, 0.059)	(-0.055, 0.080)
RF with GS TFs	(0.136, 0.437)	(0.088, 0.261)	-	(-0.290, -0.122)	(0.032, 0.266)	(0.072, 0.303)
Linear with GS TFs	(0.316, 0.669)	(0.259, 0.503)	(0.122, 0.290)	-	(0.203, 0.508)	(0.248, 0.539)
RF with top TFs	(0.074, 0.200)	(-0.059, 0.110)	(-0.266, -0.032)	(-0.508, -0.203)	-	(-0.025, 0.101)
RF with minimal TF set	(0.029, 0.169)	(-0.080, 0.055)	(-0.303, -0.072)	(-0.539, -0.248)	(-0.101, 0.025)	-

Table 3.7: Human: Paired non-parametric results for the performance comparison between the model in blue and orange measured using root mean square error (RMSE) on target gene expressions in the test datasets. Entry (i, j) shows the 95% confidence interval as the difference of the ith blue modeling method minus the jth orange modeling method. When the difference in the blue method i with the orange method j has a p-value below 0.05 based on a non-parametric paired test, the (i, j)th confidence interval will be colored accordingly. A negative number means the method in blue has a lower error so is better, and vice versa. Otherwise the (i, j)th entry will be black. Please see the caption of Table 3.3 for a glossary of terms.

	RF with all TFs	Linear with all TFs	RF with GS TFs	Linear with GS TFs	RF with top TFs	RF with minimal TF set
Mean RMSE	0.931	1.088	1.049	1.474	0.995	1.004
RF with all TFs	-	(-0.192, -0.122)	(-0.143, -0.093)	(-0.616, -0.469)	(-0.081, -0.047)	(-0.088, -0.056)
Linear with all TFs	(0.122, 0.192)	-	(0.005, 0.073)	(-0.457, -0.314)	(0.060, 0.126)	(0.052, 0.117)
RF with GS TFs	(0.093, 0.143)	(-0.073, -0.005)	-	(-0.489, -0.361)	(0.032, 0.076)	(0.019, 0.072)
Linear with GS TFs	(0.469, 0.616)	(0.314, 0.457)	(0.361, 0.489)	-	(0.413, 0.545)	(0.400, 0.541)
RF with top TFs	(0.047, 0.081)	(-0.126, -0.060)	(-0.076, -0.032)	(-0.545, -0.413)	-	(-0.029, 0.012)
RF with minimal TF set	(0.056, 0.088)	(-0.117, -0.052)	(-0.072, -0.019)	(-0.541, -0.400)	(-0.012, 0.029)	-

side effects such manipulation might have on other genes. This is a direction for future work.

3.4.2 Batch Effects

While the z-score takes care of quantity bias in different tests, batch effects may cause predictions on batch A based on data from batch A to be superior to predictions on batch A from data on many batches. That is a limitation of any predictive model in biology.

To test this, we created our models based on multiple batches and tested them on the tail end of all those batches. We compared that approach with batch-by-batch predictions. Figure 3.5 shows that the same model trained on all batches of data achieves the same level or better predictive performance compared to using batch X's data on batch X's tail, for each batch X.

3.4.3 Ensemble of Disjoint Sets of Transcription Factors

Another potential use of the identification of disjoint sets of predictive TFs stems from the fact that each disjoint set represents a regression model for the prediction of the target gene. For all the disjoint sets we found of a target gene g, the regression model of each disjoint set can give a prediction about the expression of the target gene given the expression input of the TFs at the previous time point. As has been shown in many past studies in both general machine learning and gene network inference [Dietterich 2000; Marbach et al. 2012; Sagi and Rokach 2018; Ganaie et al. 2022], an ensemble consisting of the arithmetic mean of these model predictions may lead to an overall better performing prediction. Inspired by those results, we compared the predictive performance of this ensemble of disjoint sets of TFs to all other RF-based regressions we discussed before, the results are presented in Table 3.8. In most cases, this ensemble yielded regression accuracies second only to the model that takes all TFs as input.

For a complete list of all the minimal sets and disjoint sets of TFs for each target gene we surveyed in this study, please refer to supplementary tables S1 through S5, which are deposited in

Table 3.8: Paired non-parametric results for the performance comparison between the model in blue and orange measured using root mean square error (RMSE) on target gene expressions in the test datasets. Each column is a comparison for one of the four species that compared the ensemble prediction from disjoint sets of transcription factors (TFs) to other random forest (RF) based methods. Entry (i, j) shows the 95% confidence interval as the difference of the ith blue modeling method minus the jth orange modeling method. When the difference in the blue method i with the orange method j has a p-value below 0.05 based on a non-parametric paired test, the (i, j)th confidence interval will be colored accordingly. A negative number means the method in blue has a lower error so is better, and vice versa. Otherwise the (i, j)th entry will be black. Please see the caption of Table 3.3 for a glossary of terms.

	Ensemble of Disjoint Sets in Yeast	Ensemble of Disjoint Sets in B.subtilis	Ensemble of Disjoint Sets in Arabidopsis	Ensemble of Disjoint Sets in Mouse	Ensemble of Disjoint Sets in Human
RF with all TFs	(-0.058, -0.017)	(-0.009, 0.008)	(-0.009, 0.001)	(-0.038, 0.013)	(-0.021, -0.003)
RF with GS TFs	(0.227, 0.366)	(-0.003, 0.034)	(0.270, 0.349)	(0.034, 0.223)	(0.115, 0.181)
RF with top TFs	(-0.010, 0.027)	(0.122, 0.170)	(0.090, 0.133)	(0.169, 0.444)	(0.085, 0.134)
RF with minimal TF set	(0.138, 0.231)	(0.196, 0.249)	(0.105, 0.146)	(0.350, 0.681)	(0.464, 0.610)

this following Google Drive link .

3.4.4 An Application: optimizing gene expression

Suppose our goal is to cause a gene g to be expressed at a certain level. We've seen that the gold standard network, even when available, gives quite poor predictions. A better approach is to start with a good predictive model for g on a small number of TFs T and then to determine values of the TFs in T that might lead to the desired expression level of g. This goal is supported by the three goals of our framework: find a good model, reduce the number of TFs while preserving accuracy, and find possible alternative sets of TFs that also give high prediction accuracy. Gene Regulatory Networks do not give natural guidance for any goal like this.

Thus, the Bipartite Network approach provides an actionable approach to causality. Along the way, it provides (i) a visualization that shows alternative ways to manipulate a target gene, and (ii) a simple ensemble approach to prediction.

3.5 Empirical findings

The following are our empirical findings:

- We confirm previous observations [Pratapa et al. 2020; Zhao et al. 2021] that non-linear models generally yield better results (as measured by Root Mean Squared Error) than linear ones.
- Using all TFs yields better predictive results than using the TFs from Gold Standard edges. For each target gene g, there often exist several disjoint minimal sets (mostly of size eight or less) that give predictive accuracy nearly as high as all TFs.
- Using all batches of each species together for training yields results on the time series test tails of each batch that are as good as or better than using each batch on its own test tail.
- For a given target gene g, forming a model consisting of the most influential k_g Transcription Factors in a non-linear model (e.g. random forests) as measured on the training set, where k_g is the number of TFs in the gold standard network that point to g, yields better prediction accuracy on the test set than using the same kind of model on the gold standard TFs. This superiority holds for all the species we've tested from yeast with a mean value of 53 TFs for each target gene to B. subtilis with a mean value of 1.9 TFs for each target gene.

3.6 Conclusion

Based on our empirical findings, we suggest a framework for studying causality in gene regulation having three main features.

First, the figure of merit for causality should be predictive accuracy rather than conformance with "gold standard" edges. One reason is epistemic: any causal model should be predictive.

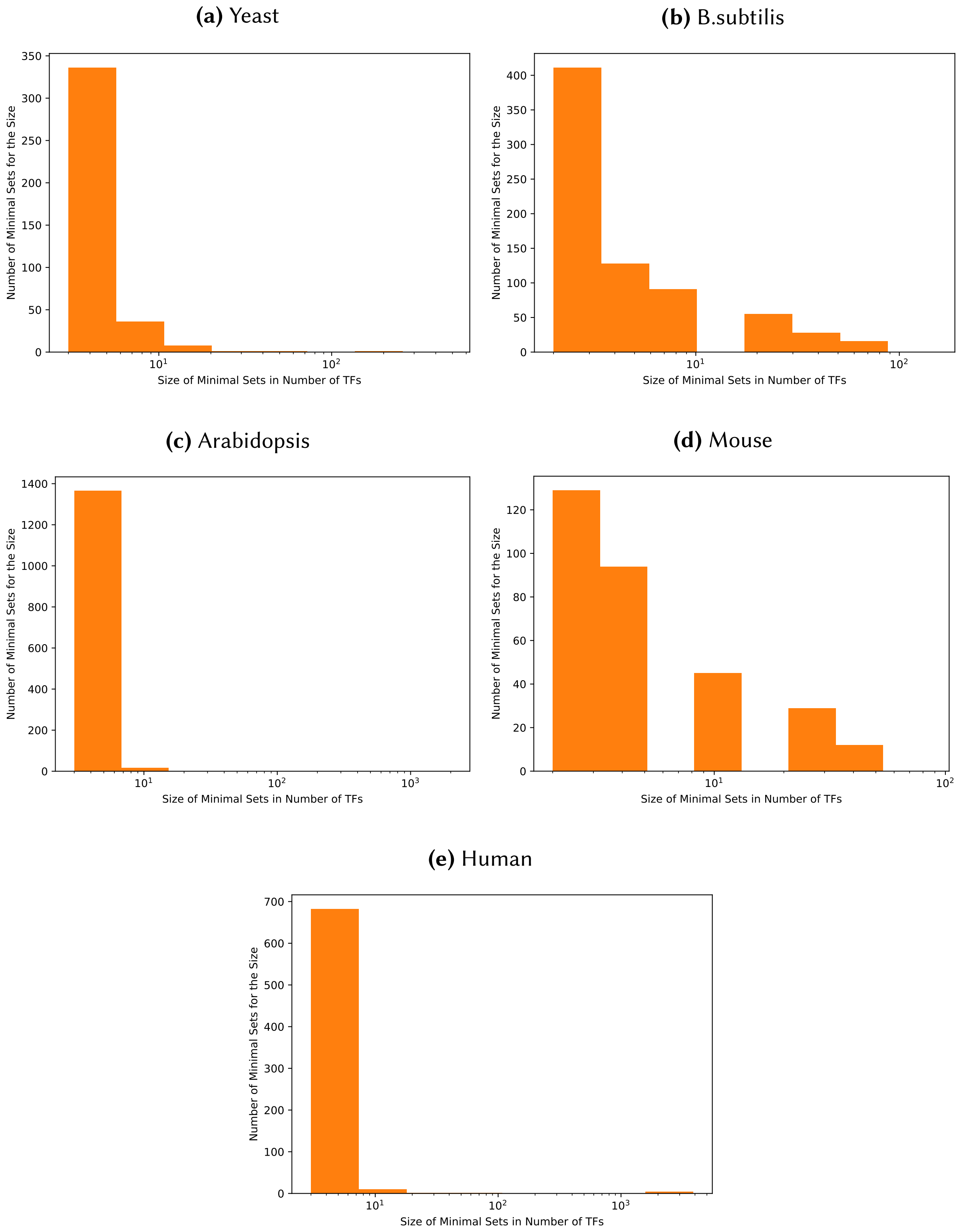
Another reason is pragmatic: prediction is useful if we want to manipulate some property such as the expression of a target gene.

Second, the network representation of such causality should be a bipartite graph consisting of gene (including transcription factor) nodes and model nodes. Such graphs encode the synergy of multiple transcription factors in the model nodes.

Third, the bipartite representation may include many model nodes that point to the same target gene, where each model node has a disjoint set of transcription factors as input. A single transcription factor plays a role in disjoint sets of several target genes.

In addition to suggesting a modified approach to causality research for transcriptional regulation, we assert that our framework applies beyond transcriptional causality. Our main future work is to apply this form of analysis to other multifactor causality domains. We welcome other researchers to try this approach and offer our software to help.

Figure 3.1: Size distribution of minimal TF set for each target gene. For every species, the majority of genes are best predicted by under eight transcription factors. That is 96.6% for yeast, 73.5% for b.subtilis, 98.7% for arabidopsis, 71.9% for mouse and 97.7% for human.



(a) Yeast

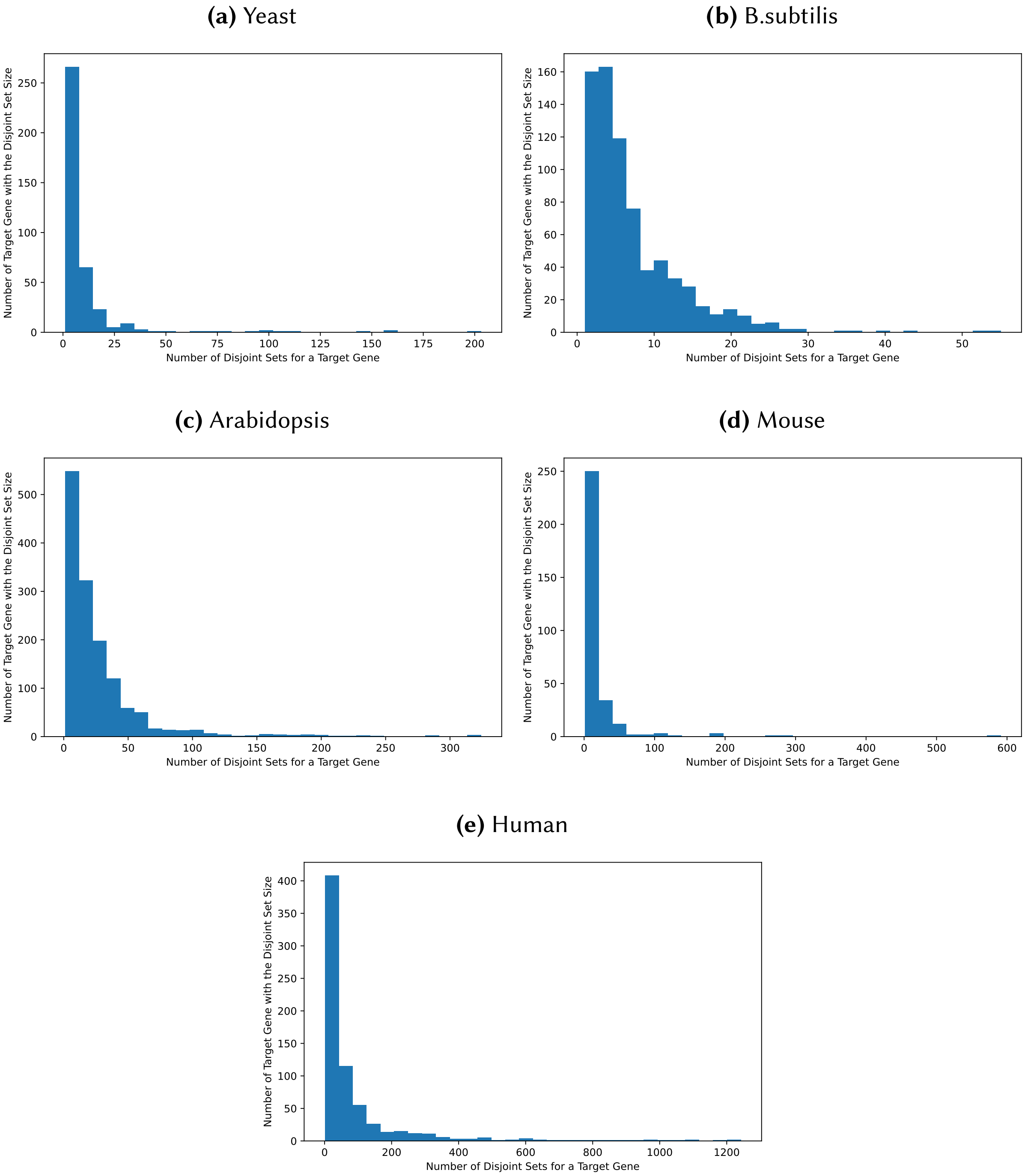
(b) B.subtilis

(c) Arabidopsis

(d) Mouse

(e) Human

Figure 3.2: Distribution of disjoint set count for each target gene. Many target genes can be best modeled by a handful of explanatory set of TFs. For yeast, 44.2% of the target genes are best modeled by fewer than five explanatory disjoint sets of TFs. For B. Subtilis, 44.1%. For Arabidopsis, 15.2%. For Mouse, 25.5%. For Human, 10.3%. Sometimes many disjoint sets of TFs are redundant. Bipartite graphs capture this causality information, as we will see below.



(a) Yeast

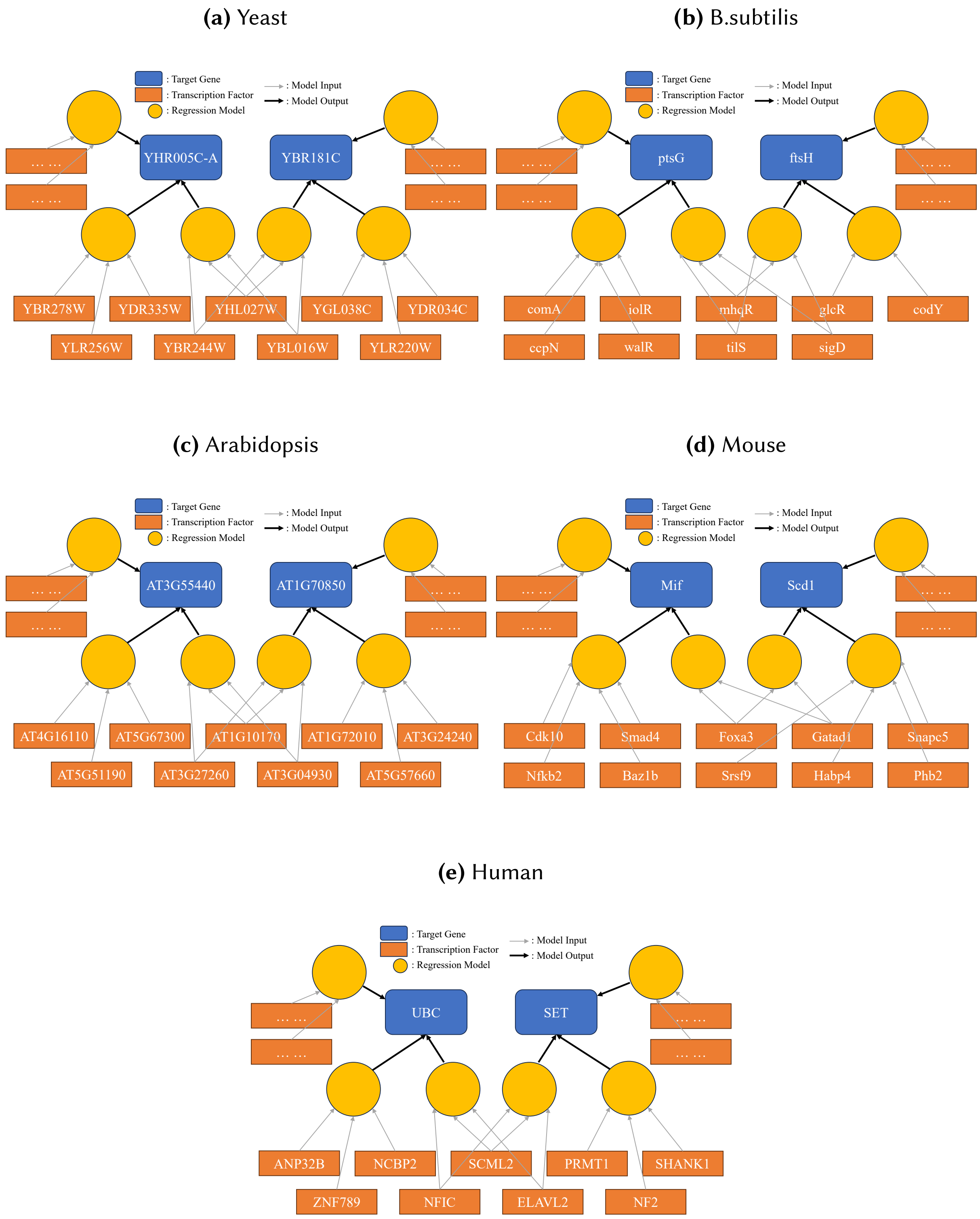
(b) B.subtilis

(c) Arabidopsis

(d) Mouse

(e) Human

Figure 3.3: Bipartite representation of causality. Light circular orange nodes represent non-linear models that take transcription factors (dark orange rectangles) as inputs and produce predictions on a single target gene (in blue). Here we show a particular case where disjoint sets of TFs can form high quality prediction models for one target gene and the same TF can be in models for several target genes.



(a) Yeast

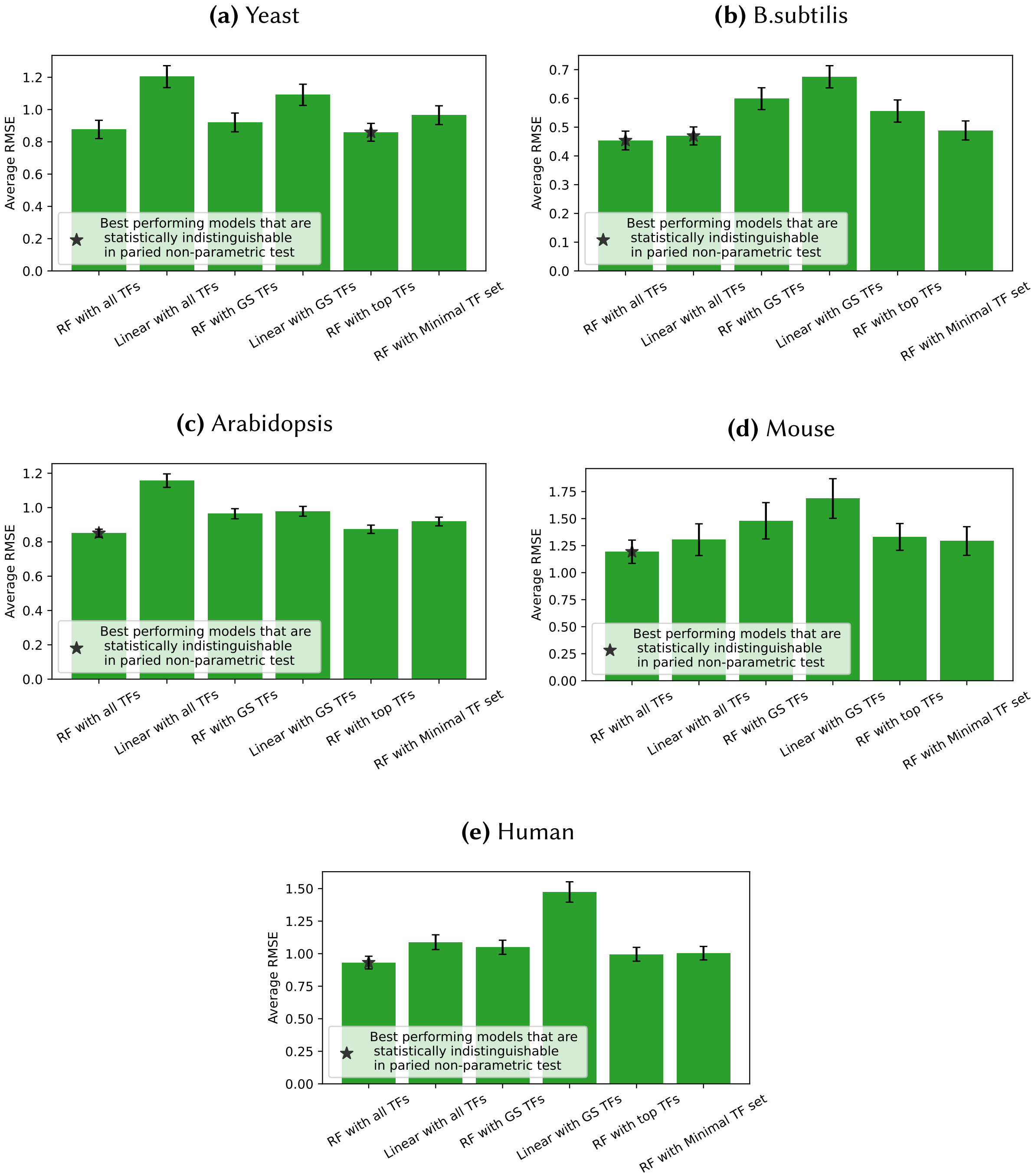
(b) B.subtilis

(c) Arabidopsis

(d) Mouse

(e) Human

Figure 3.4: Root Mean Squared Error (RMSE) performance (lower is better) of different regression models compared across four species, error bars representing the standard error of each group. When we compare all the models for each of the tested target genes, a paired non-parametric test can be applied between each pair of models to see if the performance is statistically different. The best performing models that are statistically indistinguishable this way are marked with $*$. “RF with all TFs" is always one of the best performing models. Please see the caption of Table 3.3 for a glossary of terms.



(a) Yeast

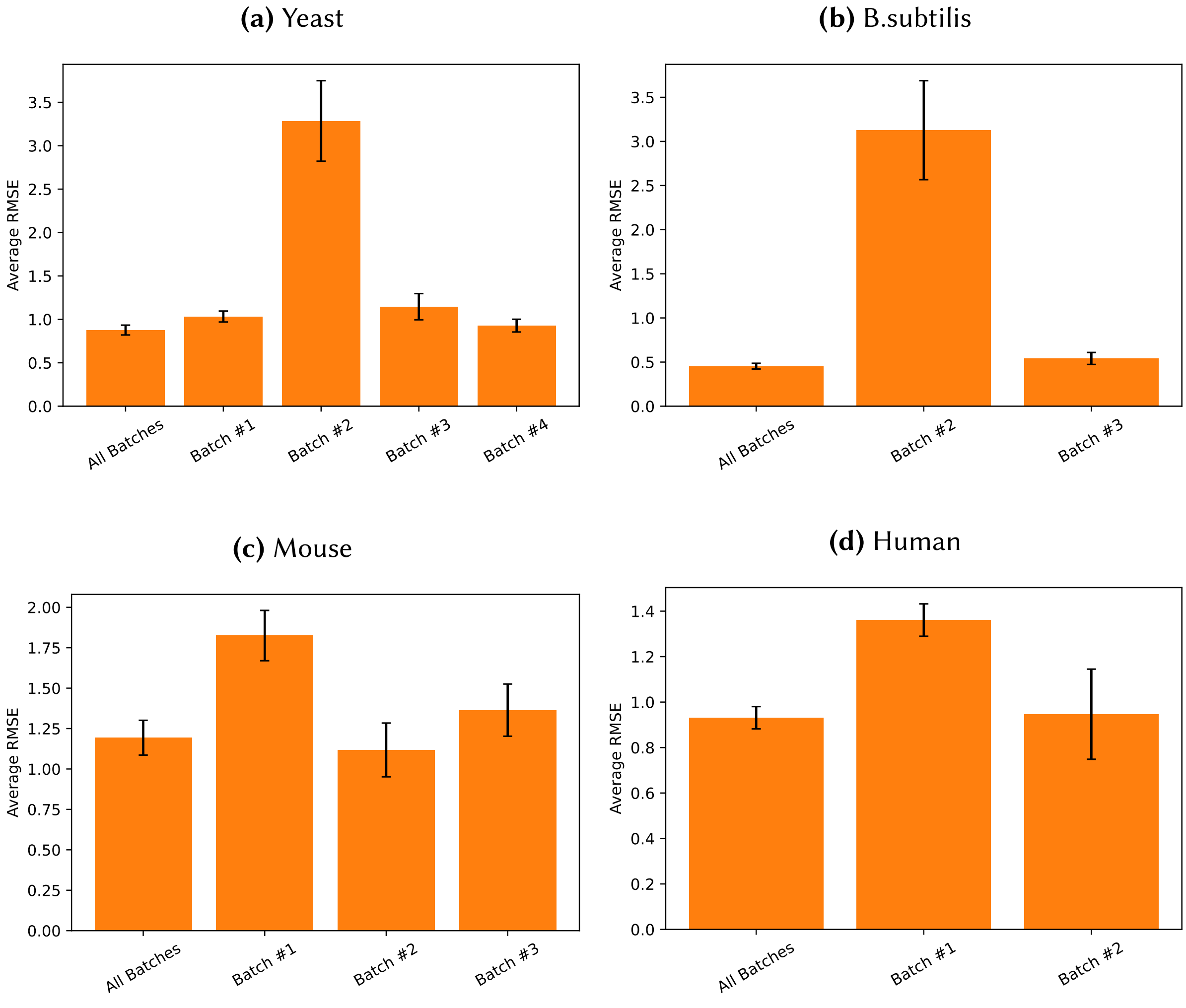
(b) B.subtilis

(c) Arabidopsis

(d) Mouse

(e) Human

Figure 3.5: Random forest regression performance differences measured across different data batches for different species. Here all batches results are the default and presented relists shown in this work where random forest model using all TFs were trained on training data from all batches and tested on all the tail testing parts from different batches. The same model was then trained and tested on individual batches for their respective high variance target genes. Note that for arabidopsis, one singular batch was used, so no need for such comparison. For b.subtilis, the first batch does not have high variance target genes in its testing set so the comparison was also omitted.



(a) Yeast

(b) B.subtilis

(c) Mouse

(d) Human

4 Discussion

This thesis presents two contributions to the field of network casual inference in biological systems, with the potential and ability to extend to other similar systems. First, we introduce *EnsInfer*, a robust ensemble approach that combines multiple network inference methods using a Naive Bayes model. Our findings demonstrate that this approach often outperforms individual inference methods across various datasets, particularly in sparse data environments. Second, we propose a novel framework *bipartite network* for studying causality in gene regulation, emphasizing predictive accuracy, bipartite graph representation, and the consideration of multiple disjoint sets of transcription factors. These contributions not only enhance our ability to infer gene regulatory networks but also offer a new perspective on representing and analyzing causal relationships in complex biological systems.

The following discussion elaborates on these two contributions, their implications, limitations, and potential for broader applications beyond transcriptional regulation.

4.1 Effectiveness of Ensemble Methods in Inference

The effectiveness of ensemble methods in network inference, particularly our *EnsInfer* approach, addresses a key challenge in the field. As noted by Pratapa et al. [Pratapa et al. 2020], no single inference method consistently outperforms others across all datasets. *EnsInfer* reflects this insight by combining multiple inference methods using a Naive Bayes model. This approach proves particularly effective in sparse data environments, which are common in biological studies [Chen and Guestrin 2016]. The simplicity of the Naive Bayes model often leads to robust performance improvements compared to more complex models in practice [Rish et al. 2001]. Our results demonstrate that *EnsInfer* frequently surpasses individual methods in predictive accuracy [Shen et al. 2023], aligning with the growing recognition of ensemble approaches in bioinformatics [Marbach et al. 2012]. Besides its robust improvement over individual inference algorithm shown even in the most challenging cases. The way *EnsInfer* is constructed also makes it inherently flexible to leverage future network inference approaches by utilizing their collective wisdom. Our flexible framework promises to push the boundaries of our understanding of biological systems and pave the way for new discoveries in diverse domains, including disease biology, developmental processes, and synthetic biology.

4.2 Limitations and Considerations of *EnsInfer*

While *EnsInfer* demonstrates significant advantages, it's important to acknowledge its limitations and considerations. In some cases, particularly when individual inference methods perform only marginally better than random, the ensemble approach may not yield substantial improvements [Marbach et al. 2012]. This was observed in single-cell human embryonic stem cell data, where base methods in BEELINE barely outperformed random predictions [Pratapa et al. 2020]. Additionally, the effectiveness of *EnsInfer* can be influenced by data sparsity and feature space size. In sparse data environments, common in experimental biology, Naive Bayes tends to perform well [Rish et al. 2001]. However, as the feature space expands with the inclusion of more inference algorithms, feature selection might become necessary to maintain performance [Guyon and Elisseeff 2003].

4.3 Proposed Framework *Bipartite Network*

Our proposed framework for studying causality in gene regulation represents a significant shift from traditional approaches. First, we advocate for using predictive accuracy as the primary figure of merit for causality, rather than conformance with "gold standard" edges. This aligns with the fundamental principle that causal models should be predictive [Pearl 2009]. Moreover, it offers practical advantages for researchers aiming to manipulate gene expression or other biological properties [Alon 2019]. This approach is particularly relevant in the context of recent advancements in single-cell genomics, where traditional bulk methods may fall short of capturing the complexity of gene regulation [Wagner et al. 2016]. By focusing on predictive power, our framework can potentially reveal causal relationships that might be overlooked by conventional methods, especially in cases where gold standards are incomplete or biased [Schaffter et al. 2011].

Second, we propose representing causality using a bipartite graph structure, consisting of gene nodes (including transcription factors) and model nodes. This representation offers several advantages over traditional network models. It allows for the explicit encoding of synergistic effects among multiple transcription factors within the model nodes, capturing complex regulatory interactions that are often oversimplified in standard graphs [Davidson 2010]. Furthermore, our framework accommodates multiple model nodes pointing to the same target gene, each with a disjoint set of transcription factors as input. This feature enables the representation of diverse regulatory mechanisms for a single gene, reflecting the biological reality of complex gene regulation [Peter and Davidson 2015]. Concurrently, it allows for individual transcription factors to participate in multiple, disjoint sets influencing various target genes, capturing the pleiotropic nature of many regulatory factors [Boyle et al. 2017]. This bipartite representation not only enhances our ability to model complex regulatory relationships but also provides a more intuitive visualization of these interactions, potentially leading to new insights in gene regulation studies [Karlebach and Shamir 2008].

4.4 Future Directions and Broader Applications

The *Bipartite Network* framework has potential applications beyond transcriptional regulation [Shen et al. 2024]. Future work should focus on applying this approach to other multifactor causality domains in biology and beyond [Barabasi and Oltvai 2004]. For instance, it could be adapted to study metabolic networks [Nielsen 2017] or signaling pathways [Kitano 2002]. The integration of new inference methods, such as those leveraging deep learning [Eraslan et al. 2019], could further enhance the power of our ensemble approach.

We and other researchers should test and adapt this framework in their respective fields. To facilitate this, we have made our software tools available to the scientific community, aiming to foster collaborative advancements in causal inference across diverse domains. It can be found at this following GitHub repository.

Bibliography

Aburomman, A. A. and Reaz, M. B. I. (2017). A survey of intrusion detection systems based on ensemble and hybrid classifiers. *Computers & security*, 65:135–152.

Alon, U. (2019). *An introduction to systems biology: design principles of biological circuits*. Chapman and Hall/CRC.

Androulakis, I. P., Yang, E., and Almon, R. R. (2007). Analysis of time-series gene expression data: methods, challenges, and opportunities. *Annu. Rev. Biomed. Eng.*, 9:205–228.

Arrieta-Ortiz, M. L., Hafemeister, C., Bate, A. R., Chu, T., Greenfield, A., Shuster, B., Barry, S. N., Gallitto, M., Liu, B., Kacmarczyk, T., et al. (2015). An experimentally supported model of the bacillus subtilis global transcriptional regulatory network. *Molecular systems biology*, 11(11):839.

Artime, O. and De Domenico, M. (2022). From the origin of life to pandemics: Emergent phenomena in complex systems.

Aubin-Frankowski, P.-C. and Vert, J.-P. (2020). Gene regulation inference from single-cell rna-seq data with linear differential equations and velocity inference. *Bioinformatics*, 36(18):4774–4780.

Aviram, R., Dandavate, V., Manella, G., Golik, M., and Asher, G. (2021). Ultradian rhythms of akt phosphorylation and gene expression emerge in the absence of the circadian clock components per1 and per2. *PLoS biology*, 19(12):e3001492.

Bar-Joseph, Z., Gitter, A., and Simon, I. (2012). Studying and modelling dynamic biological processes using time-series gene expression data. *Nature Reviews Genetics*, 13(8):552–564.

Barabasi, A.-L. and Oltvai, Z. N. (2004). Network biology: understanding the cell's functional organization. *Nature reviews genetics*, 5(2):101–113.

Bonneau, R., Reiss, D. J., Shannon, P., Facciotti, M., Hood, L., Baliga, N. S., and Thorsson, V. (2006). The inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome biology*, 7(5):1–16.

Boyle, E. A., Li, Y. I., and Pritchard, J. K. (2017). An expanded view of complex traits: from polygenic to omnigenic. *Cell*, 169(7):1177–1186.

Brooks, M. D., Juang, C.-L., Katari, M. S., Alvarez, J. M., Pasquino, A., Shih, H.-J., Huang, J., Shanks, C., Cirrone, J., and Coruzzi, G. M. (2021). Connectf: a platform to integrate transcription factor–gene interactions and validate regulatory networks. *Plant physiology*, 185(1):49–66.

Cazarin, J., DeRollo, R. E., Shahidan, S. N. A. B. A., Burchett, J. B., Mwangi, D., Krishnaiah, S., Hsieh, A. L., Walton, Z. E., Brooks, R., Mello, S. S., et al. (2023). Myc disrupts transcriptional and metabolic circadian oscillations in cancer and promotes enhanced biosynthesis. *PLoS genetics*, 19(8):e1010904.

Chan, T. E., Stumpf, M. P., and Babtie, A. C. (2017). Gene regulatory network inference from single-cell data using multivariate information measures. *Cell systems*, 5(3):251–267.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Chu, L.-F., Leng, N., Zhang, J., Hou, Z., Mamott, D., Vereide, D. T., Choi, J., Kendziorski, C., Stewart, R., and Thomson, J. A. (2016). Single-cell rna-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm. *Genome biology*, 17(1):1–20.

Cirrone, J., Brooks, M. D., Bonneau, R., Coruzzi, G. M., and Shasha, D. E. (2020). Outpredict: multiple datasets can improve prediction of expression and inference of causality. *Scientific reports*, 10(1):1–9.

Davidson, E. H. (2010). *The regulatory genome: gene regulatory networks in development and evolution*. Elsevier.

Delgado, F. M. and Gómez-Vela, F. (2019). Computational methods for gene regulatory networks reconstruction and analysis: A review. *Artificial intelligence in medicine*, 95:133–145.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer.

Efron, B. (2020). Prediction, estimation, and attribution. *International Statistical Review*, 88:S28–S59.

Emmert-Streib, F., Dehmer, M., and Haibe-Kains, B. (2014). Gene regulatory networks and their applications: understanding biological and medical problems in terms of networks. *Frontiers in cell and developmental biology*, 2:38.

Eraslan, G., Avsec, Ž., Gagneur, J., and Theis, F. J. (2019). Deep learning: new computational modelling techniques for genomics. *Nature Reviews Genetics*, 20(7):389–403.

Feder, Z. A., Ali, A., Singh, A., Krakowiak, J., Zheng, X., Bindokas, V. P., Wolfgeher, D., Kron, S. J., and Pincus, D. (2021). Subcellular localization of the j-protein sis1 regulates the heat shock response. *Journal of Cell Biology*, 220(1).

Ganaie, M. A., Hu, M., Malik, A., Tanveer, M., and Suganthan, P. (2022). Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115:105151.

Gibbs, C. S., Jackson, C. A., Saldi, G.-A., Shah, A., Tjärnberg, A., Watters, A., De Veaux, N., Tchourine, K., Yi, R., Hamamsy, T., et al. (2021). Single-cell gene regulatory network inference at scale: The inferelator 3.0. *bioRxiv*.

Graham, D. B., Jasso, G. J., Mok, A., Goel, G., Ng, A. C., Kolde, R., Varma, M., Doench, J. G., Root, D. E., Clish, C. B., et al. (2018). Nitric oxide engages an anti-inflammatory feedback loop mediated by peroxiredoxin 5 in phagocytes. *Cell reports*, 24(4):838–850.

Greenfield, A., Madar, A., Ostrer, H., and Bonneau, R. (2010). Dream4: Combining genetic and dynamic information to identify biological networks and dynamical models. *PloS one*, 5(10):e13397.

Greenwell, B. J., Beytebiere, J. R., Lamb, T. M., Bell-Pedersen, D., Merlin, C., and Menet, J. S. (2020). Isoform-specific regulation of rhythmic gene expression by alternative polyadenylation. *bioRxiv*, pages 2020–12.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.

Harris, A. and Ünal, E. (2023). The transcriptional regulator ume6 is a major driver of early gene expression during gametogenesis. *Genetics*, 225(2):iyad123.

Haury, A.-C., Mordelet, F., Vera-Licona, P., and Vert, J.-P. (2012). Tigress: trustful inference of gene regulation using stability selection. *BMC systems biology*, 6(1):1–17.

Hayashi, T., Ozaki, H., Sasagawa, Y., Umeda, M., Danno, H., and Nikaido, I. (2018). Single-cell full-length total rna sequencing uncovers dynamics of recursive splicing and enhancer rnas. *Nature communications*, 9(1):1–16.

Heerah, S., Molinari, R., Guerrier, S., and Marshall-Colon, A. (2021). Granger-causal testing for irregularly sampled time series with application to nitrogen signalling in arabidopsis. *Bioinformatics*, 37(16):2450–2460.

Hill, S. M., Heiser, L. M., Cokelaer, T., Unger, M., Nesser, N. K., Carlin, D. E., Zhang, Y., Sokolov, A., Paull, E. O., Wong, C. K., et al. (2016). Inferring causal molecular networks: empirical assessment through a community-based effort. *Nature methods*, 13(4):310–318.

Huynh-Thu, V. A., Irrthum, A., Wehenkel, L., and Geurts, P. (2010). Inferring regulatory networks from expression data using tree-based methods. *PloS one*, 5(9):e12776.

Huynh-Thu, V. A. and Sanguinetti, G. (2019). Gene regulatory network inference: an introductory survey. *Gene regulatory networks: Methods and protocols*, pages 1–23.

Jan, Z. and Verma, B. (2020). Multiple strong and balanced cluster-based ensemble of deep learners. *Pattern Recognition*, 107:107420.

Karlebach, G. and Shamir, R. (2008). Modelling and analysis of gene regulatory networks. *Nature reviews Molecular cell biology*, 9(10):770–780.

Katari, M. S., Tyagi, S., and Shasha, D. E. (2021). *Statistics is easy: case studies on real scientific datasets*. Springer.

Kharchenko, P. V., Silberstein, L., and Scadden, D. T. (2014). Bayesian approach to single-cell differential expression analysis. *Nature methods*, 11(7):740–742.

Kim, S. (2015). ppcor: an r package for a fast calculation to semi-partial correlation coefficients. *Communications for statistical applications and methods*, 22(6):665.

Kitano, H. (2002). Systems biology: a brief overview. *science*, 295(5560):1662–1664.

Krawczyk, A. O., de Jong, A., Eijlander, R. T., Berendsen, E. M., Holsappel, S., Wells-Bennik, M. H., and Kuipers, O. P. (2015). Next-generation whole-genome sequencing of eight strains of bacillus cereus, isolated from food. *Genome Announcements*, 3(6):10–1128.

Li, Y., Hartemink, A. J., and MacAlpine, D. M. (2021). Cell-cycle–dependent chromatin dynamics at replication origins. *Genes*, 12(12):1998.

Liang, S., Fuhrman, S., Somogyi, R., et al. (1998). Reveal, a general reverse engineering algorithm for inference of genetic network architectures. In *Pacific symposium on biocomputing*, volume 3, pages 18–29.

Liu, Z.-P., Wu, C., Miao, H., and Wu, H. (2015). Regnetwork: an integrated database of transcriptional and post-transcriptional regulatory networks in human and mouse. *Database*, 2015:bav095.

Madhamshettiwar, P. B., Maetschke, S. R., Davis, M. J., Reverter, A., and Ragan, M. A. (2012). Gene regulatory network inference: evaluation and application to ovarian cancer allows the prioritization of drug targets. *Genome medicine*, 4:1–16.

Marbach, D., Costello, J. C., Küffner, R., Vega, N. M., Prill, R. J., Camacho, D. M., Allison, K. R., Kellis, M., Collins, J. J., et al. (2012). Wisdom of crowds for robust gene network inference. *Nature methods*, 9(8):796–804.

Matsumoto, H., Kiryu, H., Furusawa, C., Ko, M. S., Ko, S. B., Gouda, N., Hayashi, T., and Nikaido, I. (2017). Scode: an efficient regulatory network inference algorithm from single-cell rna-seq during differentiation. *Bioinformatics*, 33(15):2314–2321.

Mitra, S., Zhong, J., Tran, T. Q., MacAlpine, D. M., and Hartemink, A. J. (2021). Robocop: jointly computing chromatin occupancy profiles for numerous factors from chromatin accessibility data. *Nucleic Acids Research*, 49(14):7925–7938.

Moerman, T., Aibar Santos, S., Bravo González-Blas, C., Simm, J., Moreau, Y., Aerts, J., and Aerts, S. (2019). Grnboost2 and arboreto: efficient and scalable inference of gene regulatory networks. *Bioinformatics*, 35(12):2159–2161.

Nielsen, J. (2017). Systems biology of metabolism. *Annual review of biochemistry*, 86:245–275.

Papili Gao, N., Ud-Dean, S. M., Gandrillon, O., and Gunawan, R. (2018). Sincerities: inferring gene regulatory networks from time-stamped single cell transcriptional expression profiles. *Bioinformatics*, 34(2):258–266.

Pearl, J. (2009). *Causality*. Cambridge university press.

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

Pedreira, T., Elfmann, C., and Stülke, J. (2022). The current state of subti wiki, the database for the model organism bacillus subtilis. *Nucleic Acids Research*, 50(D1):D875–D882.

Peter, I. S. and Davidson, E. H. (2015). *Genomic control process: development and evolution*. Academic Press.

Pisithkul, T., Schroeder, J. W., Trujillo, E. A., Yeesin, P., Stevenson, D. M., Chaiamarit, T., Coon, J. J., Wang, J. D., and Amador-Noguez, D. (2019). Metabolic remodeling during biofilm development of bacillus subtilis. *MBio*, 10(3):10–1128.

Pratapa, A., Jalihal, A. P., Law, J. N., Bharadwaj, A., and Murali, T. (2020). Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nature methods*, 17(2):147–154.

Prill, R. J., Marbach, D., Saez-Rodriguez, J., Sorger, P. K., Alexopoulos, L. G., Xue, X., Clarke, N. D., Altan-Bonnet, G., and Stolovitzky, G. (2010). Towards a rigorous assessment of systems biology models: the dream3 challenges. *PloS one*, 5(2):e9202.

Qiu, X., Rahimzamani, A., Wang, L., Mao, Q., Durham, T., McFaline-Figueroa, J. L., Saunders, L., Trapnell, C., and Kannan, S. (2018). Towards inferring causal gene regulatory networks from single cell expression measurements. *BioRxiv*, page 426981.

Reiter, F., Wienerroither, S., and Stark, A. (2017). Combinatorial function of transcription factors and cofactors. *Current opinion in genetics & development*, 43:73–81.

Rish, I. et al. (2001). An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. Citeseer.

Sagi, O. and Rokach, L. (2018). Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1249.

Saint-Antoine, M. M. and Singh, A. (2020). Network inference in systems biology: recent developments, challenges, and applications. *Current opinion in biotechnology*, 63:89–98.

Schaffter, T., Marbach, D., and Floreano, D. (2011). Genenetweaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics*, 27(16):2263–2270.

Shahabadi, M. S. E., Tabrizchi, H., Rafsanjani, M. K., Gupta, B., and Palmieri, F. (2021). A combination of clustering-based under-sampling with ensemble methods for solving imbalanced class problem in intelligent systems. *Technological Forecasting and Social Change*, 169:120796.

Shasha, D. and Wilson, M. (2010). Statistics is easy! *Synthesis Lectures on Mathematics and Statistics*, 3(1):1–174.

Shen, B., Coruzzi, G., and Shasha, D. (2023). Ensinfer: a simple ensemble approach to network inference outperforms any single method. *BMC bioinformatics*, 24(1):114.

Shen, B., Curozzi, G., and Shasha, D. (2024). Bipartite networks represent causality better than simple networks: evidence, algorithms, and applications. *Frontiers in Genetics*, 15:1371607.

Shu, H., Zhou, J., Lian, Q., Li, H., Zhao, D., Zeng, J., and Ma, J. (2021). Modeling gene regulatory networks using neural network architectures. *Nature Computational Science*, 1(7):491–501.

Skok Gibbs, C., Jackson, C. A., Saldi, G.-A., Tjärnberg, A., Shah, A., Watters, A., De Veaux, N., Tchourine, K., Yi, R., Hamamsy, T., et al. (2022). High-performance single-cell gene regulatory network inference at scale: the inferelator 3.0. *Bioinformatics*, 38(9):2519–2528.

Specht, A. T. and Li, J. (2017). Leap: constructing gene co-expression networks for single-cell rna-sequencing data using pseudotime ordering. *Bioinformatics*, 33(5):764–766.

Spies, D. and Ciaudo, C. (2015). Dynamics in transcriptomics: advancements in rna-seq time course and downstream analysis. *Computational and structural biotechnology journal*, 13:469–477.

Stark, R., Grzelak, M., and Hadfield, J. (2019). Rna sequencing: the teenage years. *Nature Reviews Genetics*, 20(11):631–656.

Teixeira, M. C., Viana, R., Palma, M., Oliveira, J., Galocha, M., Mota, M. N., Couceiro, D., Pereira, M. G., Antunes, M., Costa, I. V., et al. (2023). Yeastract+: A portal for the exploitation of global transcription regulation and metabolic model data in yeast biotechnology and pathogenesis. *Nucleic Acids Research*, 51(D1):D785–D791.

Tran, T. Q., MacAlpine, H. K., Tripuraneni, V., Mitra, S., MacAlpine, D. M., and Hartemink, A. J. (2021). Linking the dynamics of chromatin occupancy and transcription with predictive models. *Genome Research*, 31(6):1035–1046.

Van de Sande, B., Flerin, C., Davie, K., De Waegeneer, M., Hulselmans, G., Aibar, S., Seurinck, R., Saelens, W., Cannoodt, R., Rouchon, Q., et al. (2020). A scalable scenic workflow for single-cell gene regulatory network analysis. *Nature Protocols*, 15(7):2247–2276.

Varala, K., Marshall-Colón, A., Cirrone, J., Brooks, M. D., Pasquino, A. V., Léran, S., Mittal, S., Rock, T. M., Edwards, M. B., Kim, G. J., et al. (2018). Temporal transcriptional logic of dynamic regulatory networks underlying nitrogen signaling and use in plants. *Proceedings of the National Academy of Sciences*, 115(25):6494–6499.

Vijesh, N., Chakrabarti, S. K., Sreekumar, J., et al. (2013). Modeling of gene regulatory networks: A review. *Journal of Biomedical Science and Engineering*, 6(02):223.

Wagner, A., Regev, A., and Yosef, N. (2016). Revealing the vectors of cellular identity with single-cell genomics. *Nature biotechnology*, 34(11):1145–1160.

Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, 5(2):241–259.

Zhao, M., He, W., Tang, J., Zou, Q., and Guo, F. (2021). A comprehensive overview and critical evaluation of gene regulatory network inference technologies. *Briefings in Bioinformatics*, 22(5):bbab009.

Zheng, R., Li, M., Chen, X., Wu, F.-X., Pan, Y., and Wang, J. (2019). Bixgboost: a scalable, flexible boosting-based method for reconstructing gene regulatory networks. *Bioinformatics*, 35(11):1893–1900.