Learning Causality in Molecular Biology

by

Jacopo Cirrone

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

New York University

January 2021

Professor Dennis Shasha

Acknowledgements

Two are the key people who have been contributing to my incredible personal and professional growth over the last few years at NYU: my advisor Dennis Shasha, and my long-time soulmate Federica Lupo. Both of them are impressive human beings and have so much in common: patience, good-heart, being able to see things clearly, and reducing any complex ideas to a simple concept.

Moreover, it would be impossible to summarize the infinite number of things I have learned or to list all the great people I have interacted with and worked with. Yet, I would like to mention the inspiring environments I have had the privilege to be part of, such as NYU Courant, NYU Data Science, NYU Bio, Simons Foundation, New York Genome Center, NYU Tandon, and in general the machine learning community in NYC.

Last, but not least, I would like to immensely thank my whole family and my friends in Italy and the US for the unconditional support.

Abstract

The Systems Biology community has invested a great deal of effort in modeling gene regulatory networks that should be able to (i) accurately predict future states and (ii) identify regulatory hubs that can be manipulated to achieve desired phenotypes. Most computational tools for the problem embody linear models (e.g. $5 * TF1 + 2 * TF2 - 0.4 * TF3$....). However, it is well known that biological interactions are highly synergistic and non-linear. Further, those tools mostly try to directly predict networks even when the discovered edges (which usually come from some assay such as Chip-seq) may have little physiological significance (e.g., may not influence gene expression).

This thesis considers an alternative approach to inferring gene causality. Specifically, we consider the problem of predicting the expression of genes at a future time point in a genomic time series. In this, we follow the philosophy that accurate prediction often corresponds to a good understanding of causality.

The prediction may rest on several sources of data: the time point immediately preceding t, the entire target time series preceding t, and ancillary data. In biology, for example, the ancillary data may consist of a network based on binding data, data from different time series, steady state data, a community-blessed gold standard network, or some combination of those. We introduce OutPredict, which is a machine learning method for time series that incorporates ancillary steady state and network data to achieve a low error in gene expression prediction. We show that OutPredict outperforms several of the best state-of-the-art methods for prediction. The predictive models OutPredict in turn generate a causal network.

Thus, this thesis presents an approach to the inference of causality based on predictions of out-of-sample time-points based on both steady state and time series

data. Because the model for each gene identifies those transcription factors that have the most importance in prediction, those important transcription factors are the most likely causal elements for that gene. We validate those predictions for a set of well-documented transcription factors in Arabidopsis. Because our methods apply to any situation in which there is time series data, ancillary data, and the need for non-linear causal models, we believe that this work will have a broad appeal to the scientific community, specifically those studying causality networks in any biological system.

Contents

List of Figures

A.9 **Novel TF regulators - CRF4, SNZ, CDF1, HHO5, HHO6 & PHL1- of NxTime genes in shoots.** A. CRF4 was chosen for initial functional validation in planta, as it responds early to N-signaling (JIT: 5min) and has a high N-specificity index. In planta ([88]) and shoot cell-based transient TF perturbation assays [13] identified 16 TFs that are regulated by NxTime and by CRF4. From this set of CRF4 targets, SNZ (JIT:10min) and CDF1 (JIT:45min) were chosen for further validation by TF perturbation in shoot cells using the TARGET system [13]. B. Genome-wide regulated targets of CRF4, SNZ and CDF1 were validated by TF over-expression in plants [1] andor shoot cells [13]. C. Independently, genome-wide regulated targets of four additional TFs in the pruned GRN - HHO56, PHL1 and TGA1- were identified in shoot cells using the TARGET assay [13]. Genome-wide regulated targets of all of these seven TFs (Panels B and C), show a significant overlap with the NxTime gene set in shoots. Six of these seven TFs show over-representation of GO terms related to the Nitrogen assimilation process. Further, four novel TFs validated here (CRF4, SNZ, CDF1 and HHO5) also show a high N-specificity of regulated genes in shoots. Note: The N-specificity listed in Panel C is estimated from the regulated genome-wide targets of these four TFs in the shoot cell TARGET assays. By contrast, the N-specificity index shown in Fig 3 was estimated from genome-wide TF-target binding (DAP-Seq) in vitro [73]. . . . 91

List of Tables

List of Appendices

Chapter 1

Introduction

State-of-the-art methods for gene regulatory network inference [52, 20, 64, 27] use machine learning on genome-wide sequencing data to predict the interactions between transcriptional regulators and target genes. A typical approach to gene network inference is to take the results of an assay, most often binding assays such as CHIP-seq, and divide the data into training and test sets. This involves excluding some of the transcription factor-target binding observations, and using the remaining training set to infer the hidden data by some method. An issue with this approach is that it presumes that the majority of binding events are physiologically meaningful, in the sense that they influence the expression of the target gene. However, it has been shown that the physiological importance of binding can be minor [32].

Another frequent issue with the paradigmatic network inference approach is that the resulting networks encode linear interactions (sum of weighted effects of causal elements). This modeling strategy makes pragmatic sense in the common situation in which the number of possible interactions is much greater than the experimental

data points, because linear models have fewer parameters to fit [33]. Unfortunately, genomic interactions are decidedly non-linear, noisy and incomplete[81].

For these reasons, we first show in Chapter 2 together with Appendix 6, A.1.8 two studies where we mainly approach the causality problem with the paradigmatic network inference [86, 16], then we discuss our prediction-based method in Chapter 3 with variants in Chapters 4 and 5. Specifically, we tackle the causality problem differently, and Chapter 3 shows the steps in our causality approach: i) Build a model for each gene g that can predict the expression of that gene in left-out time points; ii) If our model is good, then the transcription factors that most influence gene g likely constitute the causal elements for g.

The underlying belief is that accurate predictive models embody causal relationships because the soundest path is to go for predictive accuracy first, and then try to understand why [15].

1.1 Dynamic Factor Graph with Plant Model Organism Data – Chapter 2

In this chapter (together with Appendix 6) we show our work that infers the temporal transcriptional logic underlying dynamic nitrogen (N) signaling in plants [86].

Any time event e1 causes e2, e1 must be earlier than e2. For that reason, time-series transcriptome data are of particular value in learning regulatory networks that can predict network states at future/untested time-points. In this chapter we learn the temporal transcriptional logic underlying dynamic nitrogen (N) signaling in plants [86] with the paradigmatic network inference.

We used a time-based machine-learning (ML) approach to infer the TF-to-target edges in the underlying gene regulatory network (GRN).

We designed and implemented a computational pipeline which includes a machine learning method based on a State-space model (SSM) algorithm called Dynamic Factor Graphs (DFG)([58]) that models the dynamics of a sequence of data by encoding the joint likelihood of observed and hidden variables.

To experimentally test these network predictions, we used genome-wide TF perturbation data for three novel TF hubs that we validated to regulate genes involved in nitrogen use in vivo. This validation step enabled us to derive a precision cut-off for TF-target edge scores in the larger GRN of 155 TFs.

Many of the TF-target edges in this refined GRN were also supported by independent TF-target binding data, used to calculate a N-specificity index for each TF. The resulting time-based GRN revealed the temporal relationships of TFs previously validated in the N-response and connects them with new TFs in the N-response cascade.

The resulting network also provided transcriptional logic for TF-perturbations aimed at improving N-use efficiency especially at low-N input levels, an important issue for global food production in marginal soils and for sustainable agriculture. A common theme throughout this thesis is that a combination of the time-based approaches and machine learning models can be applied more generally to uncover the temporal transcriptional logic for any response system in biology, agriculture or medicine [24].

In Appendix A.1.8 we present a novel approach called Network Walking which exploits the DFG algorithm and we apply to Arabidopsis root gene data. Network

Walking combines functionally validated (85,144 edges produced by our biologist colleagues at NYU Biology) and time-inferred TF-target edges to connect TF targets validated in root cells, with indirect targets regulated in planta [16]. In our proof-of-concept Network Walking examples, we determine the network path for two known TFs in the Nitrogen response, TGA1 and CRF4. Using this approach, we connect 77% and 87% of the indirect targets detected only in planta, back to TGA1 and CRF4, respectively, through intermediate TF2s. The Network Walking approach has general application across biological systems. Our proof-of-concept examples have implications for manipulation of networks that control plant N-use efficiency, a process that impacts agriculture, the environment, and human health.

1.2 OutPredict – Chapter 3

In chapter 3 we present a new method called OutPredict. Here, we have approached the causality problem differently: we first create a model for each gene g which can predict the expression of that gene in out-of-sample time points. If our model is good, then the transcription factors most influential to gene g are likely to be the causative components for g.

The form of the model is important here. Small data sizes relative to the number of causal elements preclude the use of neural networks and, in particular, deep neural networks, which would increase the number of model's parameters. The presence of non-linear relationships excludes linear methods. As a compromise, therefore, this work uses Random Forests (RF) because they model non-linear synergistic interactions of features and perform well even when sample sizes are small [14] though noise is always an issue.

The Random Forests within our new method OutPredict (OP) consist of an ensemble of regression trees tuned through extensive bootstrap sampling. We show the following: (i) The OutPredict model allows for non-linear dependencies of target genes on causal transcription factors; (ii) OutPredict can incorporate time series, steady-state, and prior (e.g. known Transcription Factor-target interactions) information to bias the forecasts; (i) OutPredict forecasts the expression value of genes at an unseen time-point better than state-of-the-art methods, partly because of steady-state and known interaction data; and (iv) the important edges inferred from OutPredict correspond to validated edges significantly more often than other state-of-the-art methods.

We compare the OutPredict method to the state-of-the-art forecasting algorithms, such as Dynamic Genie3 [38], that support forecasting and non-linear relationships, but currently lack the ability to incorporate priors. Other time-based machine learning methods such as Inferelator [33] and Dynamic Factor Graph [58], which we used in our previous studies [16, 86] are based on regularized linear regression. We also compare OutPredict with a neural net-based method built to predict gene expression time series [82].

Another relevant time series method from the literature is Granger causality, which has been used successfully for small numbers of genes[21, 94]. Granger causality is a vector autoregressive method that can be used to infer important transcription factors. In our case, however, we are trying to optimize predictive power using a large number of candidate transcription factors using very short time series (e.g. 6 time points). As is well known[54], Granger causality can give misleading results in such a setting because the time series are short, causal relationships are non-linear, and the time series are non-stationary.

1.3 Learning with Steady-State Data Alone – Chapter 4

We apply OutPredict to a single-cells dataset in Arabidopsis where each single-cell expression profile is treated as a steady-state condition, allowing the model to learn a function that maps expression values of TFs to the expression value of each target gene in the context of single-cells expression data.

The machine learning analysis belongs to an elaborated pipeline composed of computational and experimental approaches included in a work which represents new ground in single-cell studies in plants in the detailed coordination of single-cell profiling and image analysis.

1.4 PhenoPredict – Chapter 5

In chapter 5 we describe how to build causality models for phenotypes. This is useful because if, say, some gene g is differentially expressed when, for example, plants have a positive phenotypic trait such as high yield, we want to induce g so that we can transform plants to try to achieve this desirable trait. As a secondary application, we can test young unmodified plants to predict which ones will have that trait. In this chapter, we discuss a case study in rice in which we were able to predict both biomass and yield in two-month old plants based on a model for plants that were just a few weeks old.

1.5 Road Map

The presentation of the results in this thesis is structured as follows. We start by introducing the importance of time series in gene expression data in Chapter 2 to infer the temporal transcriptional logic of dynamic regulatory networks underlying nitrogen signaling in plants shoot tissue using the *state-space learning model* called *Dynamic Factor Graph*. Then in Chapter 3 we introduce our novel method called *OutPredict* which approaches the problem of learning causality differently than the paradigmatic network inference approach.

Chapter 4 describes an application of OutPredict on pure steady-state data in the context of a single-cells dataset.

We finally present in Chapter 5 PhenoPredict which constructs causality models to predict phenotypes and then learning the effect of the features (i.e., genes and TFs) on phenotypes. We conclude the thesis in Chapter 6.

Chapter 2

Dynamic Factor Graph with Plant Model Organism

2.1 Temporal transcriptional logic of dynamic regulatory networks underlying nitrogen signaling and use in plants

To infer transcription factor (TF)-target edges in a Gene Regulatory Network (GRN), we applied a time-based machine learning method to 2,174 dynamic Nitrogen(N)-responsive genes using the paradigmatic network inference. We experimentally determined a network precision cut-off, using Transcription Factor(TF)-regulated genome-wide targets of three novel TF hubs (CRF4, SNZ, CDF1 - see Appendix 6), used to prune the network to 155 TFs and 608 targets. This network precision was reconfirmed using genome-wide TF-target regulation data for four additional TFs (TGA1, HHO5/6, PHL1 - see Appendix 6) not used in network pruning.

These higher-confidence edges in the GRN were further filtered by independent TF- target binding data, used to calculate a TF N-specificity index (see Appendix 6). This refined GRN identifies the temporal relationship of known/validated regulators of N-signaling (NLP7/8, TGA1/4, NAC4, HRS1, LBD37/38/39) and 146 novel regulators. Six novel TFs - CRF4, SNZ, CDF1, HHO5/6, PHL - validated in this study by our biologist colleagues at NYU Biology (see Appendix 6), regulate a significant number of genes in the dynamic N-response, targeting 54% N-uptake/assimilation pathway genes. Phenotypically, inducible over-expression of CRF4 in planta regulates genes resulting in altered biomass, root development and $^{15}NO_3$ - uptake, specifically under low-N conditions (see Appendix 6). This dynamic N-signaling GRN now provides the temporal transcriptional logic for 155 candidate TFs to improve Nitrogen Use Efficiency (NUE) with potential agricultural applications. Broadly, these time-based approaches can uncover the temporal transcriptional logic for any biological response system in biology, agriculture or medicine.

2.2 Inferring a time-derived GRN driving the temporal N-response in shoots

De novo network inference is a valuable approach to build GRNs ([65, 52, 90, 2]). Because causality moves forward in time, fine-scale time-series transcriptome experiments are an especially valuable resource to infer GRNs that can predict out-of-sample target gene behavior, the ultimate goal of systems biology ([48],

[47], [18]). Previously, we applied a time-based machine-learning method, Dynamic Factor Graphs (DFG)([58]), to learn and predict causal relationships between TFs and their targets ([48], [47]).

DFG identifies the likely set of TFs driving target gene expression, by learning an f function that explains the target gene expression at each time-point, based on the expression of the TFs at previous time-points ([58]).

DFG uses an Expectation-Maximization (EM) algorithm which consists of a two-steps iterative procedure: the step which infers the latent variables $z(t)$ (i.e. the inference step) and the step to learn the linear f function (i.e. the learning step) (Figure 2.1).

In the inference step the model infers the latent variable $z(t)$ which represents the denoised version of the gene expression data under the assumption that mRNA data $y(t)$ are noisy observations [57] (Figure 2.1).

For each gene j, the model relationship involves the rate of gene expression change, the kinetic time constant τ, and a linear function f represented by an NxM matrix F, where N is the total number of genes and M the number of transcription factors plus a bias term b and a Gaussian error term with zero mean and fixed covariance [57]:

$$\tau \frac{z_j(t_{i+1}) - z_j(t_i)}{t_{i+1} - t_i} + z_j(t_i) = \sum_{m=1}^{M} F_{j,m}(z_m(t_i)) + b_j + \eta_j(t_i), j \in (1, N)$$

where $z_j(t_i)$ denotes the expression of gene j at t_i; $\tau = 1/\alpha$ denotes the kinetic time constant, which is related to the half-life $t_{1/2}$ by $t_{1/2} = \tau log(2)$. All genes are assumed to have the same τ (or same α).

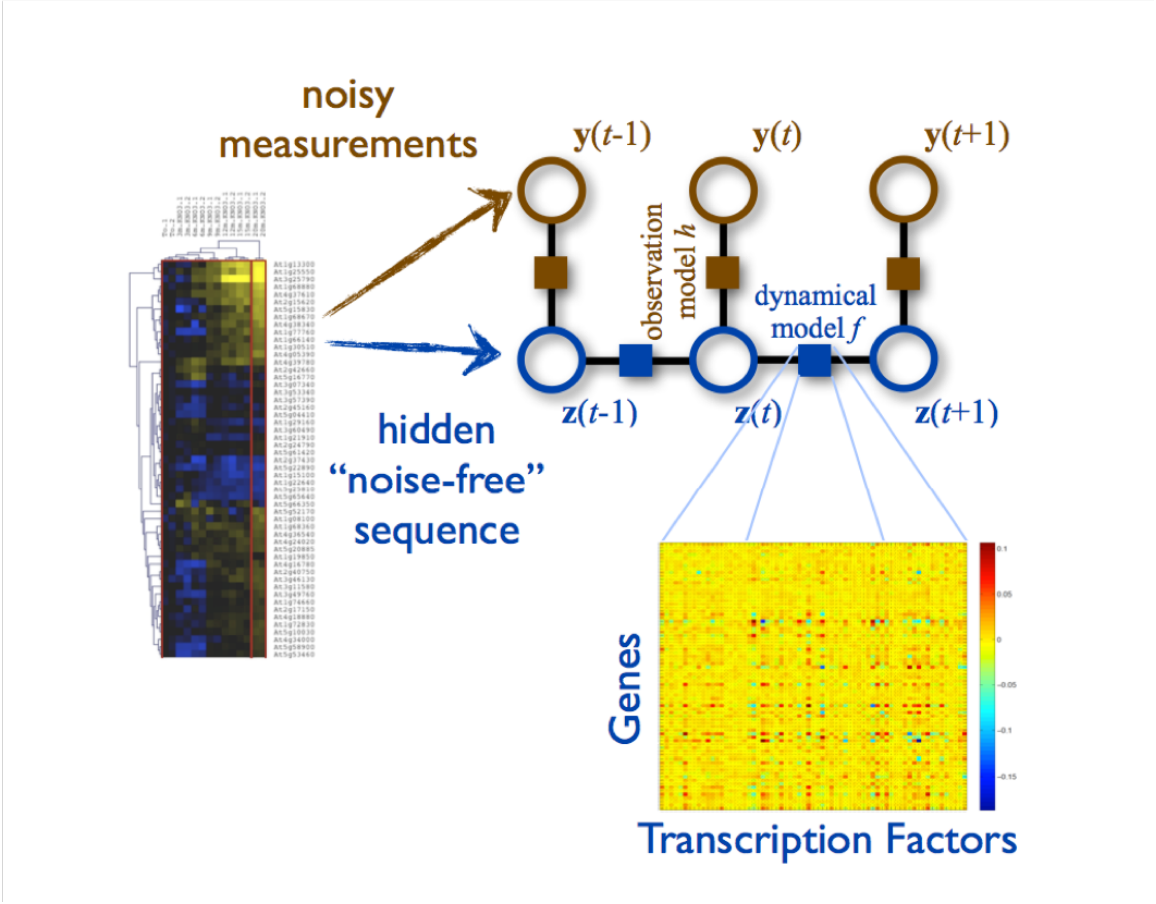


Figure 2.1: Figure from [57]. Dynamic Factor Graph representation of the state-space model learned using the time series gene expression data. The model infers the latent variable $z(t)$ which represents the denoised version of the gene expression data under the assumption that mRNA $y(t)$ are noisy observations. [57]

Here, we used the DFG method to predict the influence of every TF on every gene in the shoot NxTime gene set, implementing rigorous hyper-parameterization steps. The resultant DFG network provides a measure of the influence (i.e., TF-target edge score) of each of the 172 TFs on the 2,174 N-responsive genes in shoots (e.g. 374,000 predicted edges). However, a major challenge in de novo network

inference is the high false positive (FP) rate of TF-target predictions ([52]). We thus estimated confidence in the edges of our time-inferred DFG network by comparing the predicted TF-target edges in the GRN to experimentally validated TF targets ([52]). This method establishes the Precision (i.e., proportion of predicted TF-target edges that are real) and Recall (i.e., proportion of real TF-target edges that are predicted) of the GRN, which can then be used to prune the network to enrich for higher-confidence TF-target predictions ([52]). To implement this network pruning step, we first retained the top 10% of DFG predictions and experimentally validated the genome-wide targets of seven TF hubs in this initial DFG network. This genome-wide TF-target validation step established the Precision vs. Recall for the larger GRN of 155 TFs (Figs. 2.2 & 2.3), as detailed below.

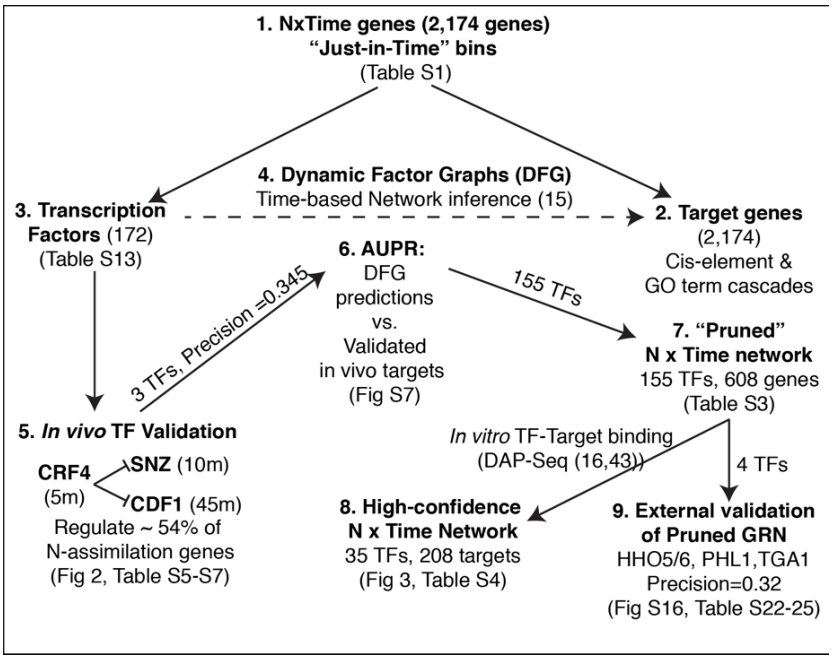


Figure 2.2: Work-flow for the time-driven de novo DFG network inference by machine learning and network pruning to increase precision of TF-target predictions. Steps 1-3. The fine-scale time-series N-response study of shoots captures 2,174 genes including 172 TFs, as per Spline analysis. Step 4. A time-based de novo inference approach called Dynamic Factor Graphs (DFG) [58, 48] was used to infer the influence of each TF on each gene in the NxTime shoot set. Step 5. Three TFs (CRF4, and its validated downstream target TFs SNZ and CDF1) that were predicted by DFG to be influential in the N-signal response GRN were selected for experimental validation by identifying their genome-wide targets in shoot cells in the TARGET system [13] or in planta[1]. Step 6. The genome-wide regulated targets of CRF4, SNZ and CDF1 were then used to compute Precision and Recall of the predicted DFG network in an Area Under the Precision Recall (AUPR) curve (Fig. 2.3). This AUPR analysis was then used to prune the network to identify a subnetwork with precision of 0.345 (Precision=0.345, corresponding TF- target edge score=0.95554) (see Fig. 2.3). Step 7. This Pruned NxTime network now connects 155 TFs to 608 N-responsive shoot genes. Step 8. Additional support for the DFG predicted edges in this pruned network was derived by overlaying available TF-target binding data from an independent source (DAP-Seq) [73, 44]. Step 9. The precision of the pruned NxTime network was independently re-validated by determining the genome-wide regulated targets of four additional TFs (HHO5/6, PHL1 and TGA1) in the GRN in shoot cells using the TARGET system [13] (this was done by experimentalists at NYU Biology). The overall precision of the predicted edges for these four new TFs is 0.32, which matches very closely the overall pruned NxTime network precision of 0.345 (Step 6).

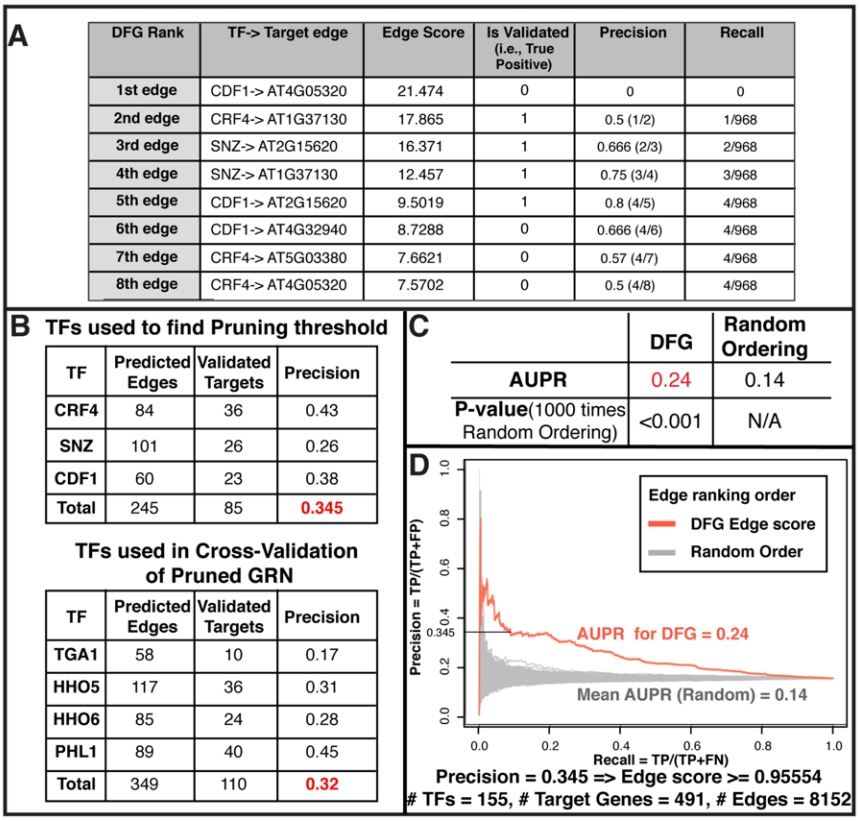


Figure 2.3: Time-inferred GRN for dynamic N-response in shoots is evaluated with validated genome-wide TF target data for Precision vs. Recall. A. Genome-wide regulated targets of CRF4, SNZ and CDF1 were compared to the ranked DFG edges in the time-based GRN to calculate Precision (True positives/(True + False positives)) and Recall (True positives/(True positive + False negative)). Some the top-ranked edge scores and validations are shown as examples. B. Genome-wide regulated targets of three TFs (CRF4, SNZ and CDF1) were used to calculate the Precision and Recall of the GRN and to prune the TF-target edges in the network to a precision threshold of 0.345. A further set of four independent TFs (TGA1, HHO5, HHO6 & PHL1) whose TF-target regulation was validated genome-wide in shoot cells, were used to cross-validate the Precision of the pruned GRN. Precision of TF-targets edges inferred for these 4 individual TFs ranges from 0.17-0.45 with an average value of 0.32. C. Area Under Precision-Recall (AUPR) analysis show that the DFG ranking of edge scores is significantly better than random order ($p < 0.001$), and the Area under PR curve (AUPR) is higher for DFG compared to random ordering ($n = 1000$). D. From the AUPR curve, the highest precision (Precision=0.345, corresponding edge score=0.9554) before the curve flattens was chosen as threshold to prune network predictions to include only higher-confidence targets.

2.2.1 Validation of novel TFs - CRF4, SNZ and CDF1 - in the time-inferred GRN that regulate N-response and N-uptake/assimilation pathway genes

To implement genome-wide validation of the DFG-predicted GRN, CRF4 was selected for initial TF-target validation as it is; i) early N-responsive in both shoots and roots (Fig. 2.5A), ii) a TF hub (422 out-edges) in the top 10% unpruned DFG shoot network (35,200 edges), iii) has a high N-specificity index (see Appendix 6), and iv) is a novel TF in N-signaling with potential links to the cytokinin pathway ([77]). Our NYU Biology colleagues and experimentalists identified the genome-wide targets of CRF4 in an inducible over-expression transplanta line (CRF4-OX)([1]), and also via TF-perturbation in shoot cells using the TARGET assay ([4, 13, 6, 41]) (see Appendix 6). These results confirm the early and central role that CRF4 plays in the dynamic N-response. In planta CRF4-regulated targets are significantly over-represented in NxTime genes in both shoots and roots, spanning early and later JIT NxTime points (Fig. 2.5B&D). The validated genome-wide targets of CRF4 in shoots include 16 downstream TFs responsive to NxTime (Fig. A.9A). We next selected two validated TF targets of CRF4 - an early (SNZ, 10 min JIT) and late (CDF1, 45 min JIT) N-responder - for TF-perturbation studies in shoot cells using the TARGET system ([13]) (Fig. A.9A&B) (this was done by experimentalists at NYU Biology). These results revealed that the targets regulated by CRF4, SNZ and CDF1 are; i. significantly enriched in NxTime genes, ii. support a high N-specificity index, and iii. are enriched in GO-terms related to Nitrate assimilation/metabolism (for CRF4, SNZ, CDF1), Ribosome biogenesis (for CRF4) and Rhythmic processes (for CDF1) (Fig. A.9B). Collectively,

the targets of CRF4, SNZ and CDF1 encompass; i) 54% of N-uptake/assimilation pathway (35/65) genes, ii) 75% of the NxTime genes in the N-uptake/assimilation pathway (12/16), and iii) 23 N-pathway genes that are not NxTime responsive (Fig. 2.4). We note the cell-based TARGET system can identify direct targets based on TF-regulation (2.4, solid lines), because translation of mRNA from primary TF-targets is blocked ([13]). By contrast, in planta TF perturbations cannot distinguish direct vs. indirect regulated targets (Fig. 2.4, dashed lines).

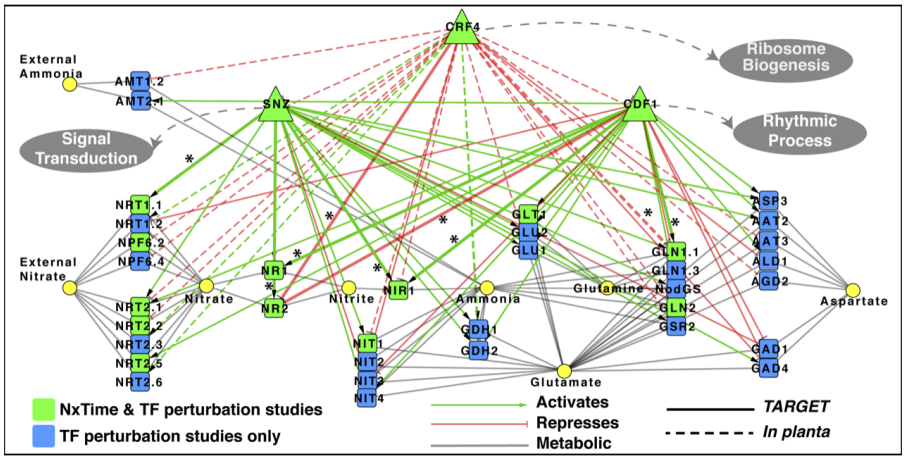


Figure 2.4: **Three novel TFs - CRF4, SNZ, CDF1 - regulate 53% of the N-uptake/assimilation pathway genes.** A time-based machine learning approach Dynamic Factor Graph (DFG) ([48], [58]) was used to infer TF-Target influence in a N-response GRN in shoots. Validated genome-wide targets of three TFs in this GRN - CRF4, SNZ and CDF1 (done by experimentalists at NYU Biology) - are shown to regulate 53% (35/65) genes in the N-uptake/assimilation pathway (Fig. A.9B). TF edges to N-responsive genes (green nodes) that are predicted by the GRN and validated by TF perturbations, are shown by (*) and thicker edge width. Grey circles indicate other cellular processes validated to be regulated by these three TFs.

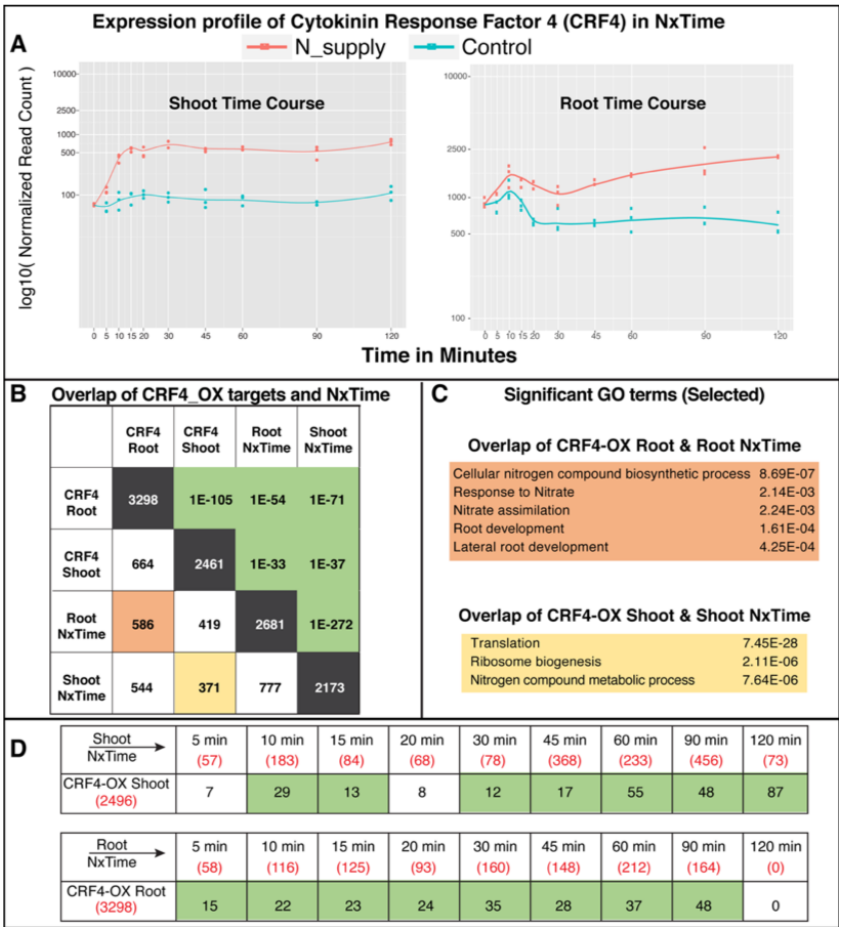


Figure 2.5: N-responsive TF hub CRF4 significantly regulates NxTime genes and N-related processes in both shoots and roots in planta. A. CRF4 responds early and robustly to N-supply in both the shoots (JIT:5mins) and roots (JIT:20mins). B. Genome-wide targets of CRF4 were determined in the shoots and roots of a conditional CRF4-OX transplanta line (24 hrs after b- Estradiol induction) [1]. The CRF4-OX regulated shoot gene set overlaps highly significantly with the Shoot NxTime gene set. The CRF4-OX regulated root gene set overlaps significantly with the Root NxTime gene set. C. GO over-representation analysis identifies that the CRF4-OX regulated targets vs. NxTime overlap in Roots is enriched for genes involved in Nitrate uptake and assimilation and genes involved in Root development which, likely, led to the observed root phenotypes in the CRF4-OX line (Fig. A.10 & Fig. A.11). In the shoots, the CRF4-OX regulated targets vs. Shoot NxTime overlap gene set was enriched in GO terms related to translational control and as well as other terms enriched in the Shoot NxTime gene set (see. Fig. A.4 and Fig. A.2D). D. Validated CRF4-OX (done by experimentalists at NYU Biology) regulated targets in the shoot and root are significantly enriched ($p<0.001$, green shading) in the early and later just- in-time gene sets in the respective organs.

2.2.2 The GRN is pruned using genome-wide TF-target validation data to identify higher-confidence edge predictions

Next, to validate our edge predictions in the time-derived GRN, we used experimentally derived TF-target regulation data for CRF4, SNZ and CDF1 (Fig. 2.2). First, we tested the significance of the DFG TF-target edge-rankings in our GRN by performing an Area Under Precision-Recall (AUPR) curve analysis ([52], [31], [50]) (Fig. 2.3). We compared the ranked TF-target edge predictions from the DFG-inferred network, to a random ranking of TF-target edges (1,000 iterations). This analysis showed that the AUPR of the DFG-inferred network (0.24), is significantly better than the mean AUPR for random networks (0.14) (p-val <0.001) (Fig. 2.3C). Next, to identify higher-confidence edges in the GRN ([31]), we chose a cut-off point (Precision $=0.345$), before the AUPR curve flattens (Fig. 2.3D). This Precision cut-off point of 0.345 matches a TF-target edge score of 0.95554 in our GRN (Fig. 2.3A&D). Thus, only TF-target edges with an edge score $>= 0.95554$ were retained in our pruned DFG network (Fig. 2.3A). This pruned GRN includes 85 validated targets out of the 245 predicted TF-target edges between CRF4, SNZ and CDF1 and genes in the NxTime shoot set. These predicted and validated targets of CRF4, SNZ and CDF1, include five key genes in N-uptake/assimilation (NRT1.1, NR1 & NR2, NIR, GLN1.1) (Fig. 2.4, edges denoted by *), ten genes involved in transcriptional/translation, and genes in the circadian clock (e.g. TIC). As our pruned GRN was optimized to increase Precision at the cost of low Recall, it likely underestimates the influence of a given TF on GRN. For example, only 9/24 experimentally validated edges from these three TFs to the 12 N-responsive

genes in the N-assimilation pathway (Fig. 2.4, green nodes) are in the pruned network (Fig. 2.4, edges with asterisk).

2.2.3 Cross-validation of the pruned N-signaling GRN using TF-target regulation data from four additional TFs

The above pruned GRN at an average precision of 0.345, can now predict the influence of 155 N-responsive TFs on 608 NxTime genes in the dynamic N-response in shoots (Fig. 2.2). To independently validate this precision rate, we identified the regulated TF-targets of four additional TFs in the GRN - HHO5/6, PHL1 and TGA1- in shoot cells using the TARGET system. The precision for each of these four TFs in the GRN ranged from 0.17 to 0.45, for an overall average of 0.32 (Fig. 2.3B & 2.7B). In total, 110/349 predicted TF-targets in the pruned GRN were experimentally validated, including six genes involved in N-uptake/reduction (Fig. 2.7A-C). These four TFs also influence a significant number of genes and processes in the NxTime gene set in shoots (Fig. A.9C). This independent TF validation proves that the initial network precision of 0.345 (Fig. 2.2) used to prune the shoot N-response GRN, extends beyond the three TFs used in the network inference pruning stage (e.g. CRF4, SNZ, CDF1), and can be used to predict targets of 155 TFs in the N-response GRN with an overall precision of 0.33. We note that our Precision cut-off of 0.33 (i.e., one in three predicted edges are likely to be true), is of comparable scale to the maximum precision of 0.5 achieved using an ensemble approach of multiple network inference methods in microbes ([52]).

2.2.4 Independent TF-target binding data supports predicted edges in the NxTime GRN

The TF-target edges in the pruned DFG network identified as hubs (i.e., influential TFs) multiple known/validated regulators of N-signaling (e.g. TGA1/4, NLP7/8, NAC4, HRS1, LBD37/38/39)([53, 42, 29, 4, 74, 26], as well as 146 potential novel regulators, including six validated herein by our NYU Biology colleagues - CRF4, SNZ, CDF1, HHO5/6, and PHL1 (Fig. A.9B&C). To add further edge support, the DFG-predicted edges in the pruned GRN were queried using an independent source of TF-target binding data for the 40 NxTime TFs in shoots that are present in the DAP-Seq dataset ([73]) (Fig. 2.6). A TF-target edge in the pruned DFG network is supported by DAP-Seq TF-target data ([73]), only if that TF is shown to bind to the promoter of the target gene in the DAP-Seq assay (41) (Fig. 2.6). We note that the actual DAP-Seq TF-DNA binding data [44] was used to establish TF-target binding, not the in silico cis-motif information ([73]). The 19 TFs in the pruned GRN that have DAP-seq data and high-N-specificity, include four known TFs in the N-response (TGA1/4, NAC4, NLP7) and four new TFs (CRF4, HHO5/6, PHL1) validated herein by experimentalists at NYU Biology (Fig. 2.6, red underlined TFs).

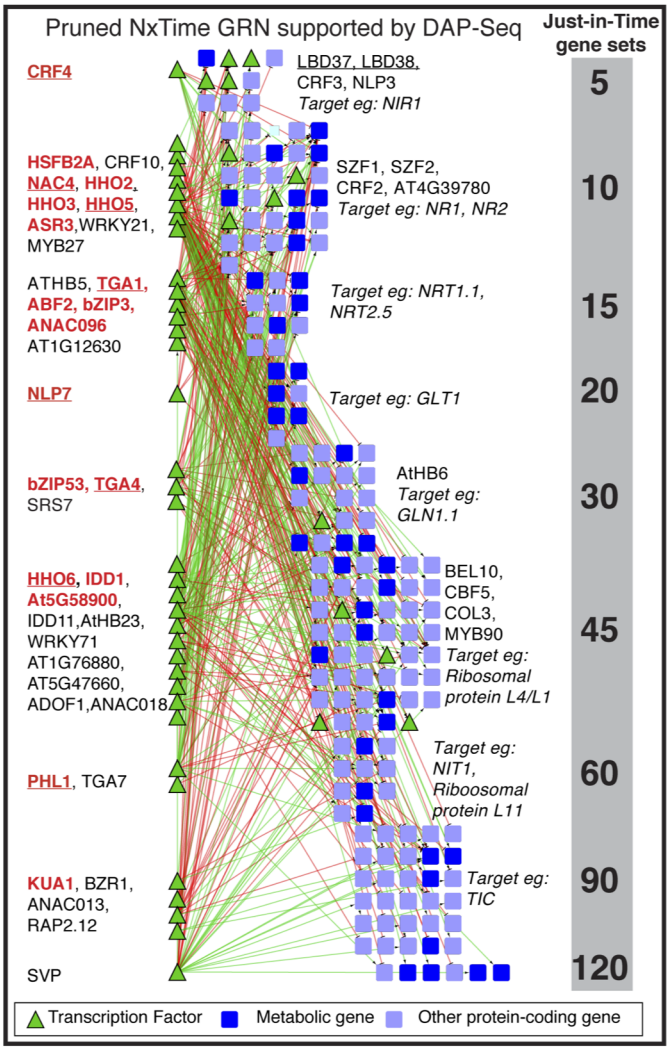


Figure 2.6: **A time-dependent GRN uncovers known and novel TFs in dynamic N-signaling in shoot.** A time- based machine learning approach Dynamic Factor Graph (DFG) ([48, 58]) was used to infer TF-Target influence in a GRN. Validated genome-wide targets of three TFs - CRF4, SNZ and CDF1 - were used to prune the GRN for TF-target Precision based on AUPR analysis (Fig. A.9 2.2 2.3). This TF-target precision was reconfirmed using data for four independent TFs - TGA1, HHO5/6, PHL1 (Fig. 2.3B). The TF-target edges supported by an independent source of TF-Target binding data (DAP-Seq ([73, 44])), capture regulation of 208 N-responsive target genes by 35 TFs. TFs with a significant N-specificity index are highlighted in red. Validated TF regulators of the N-response are underlined; NLP7(7), TGA1/4(8), NAC4(9), LBD37,38 (11), and CRF4, HHO5/6, PHL1 [This study].

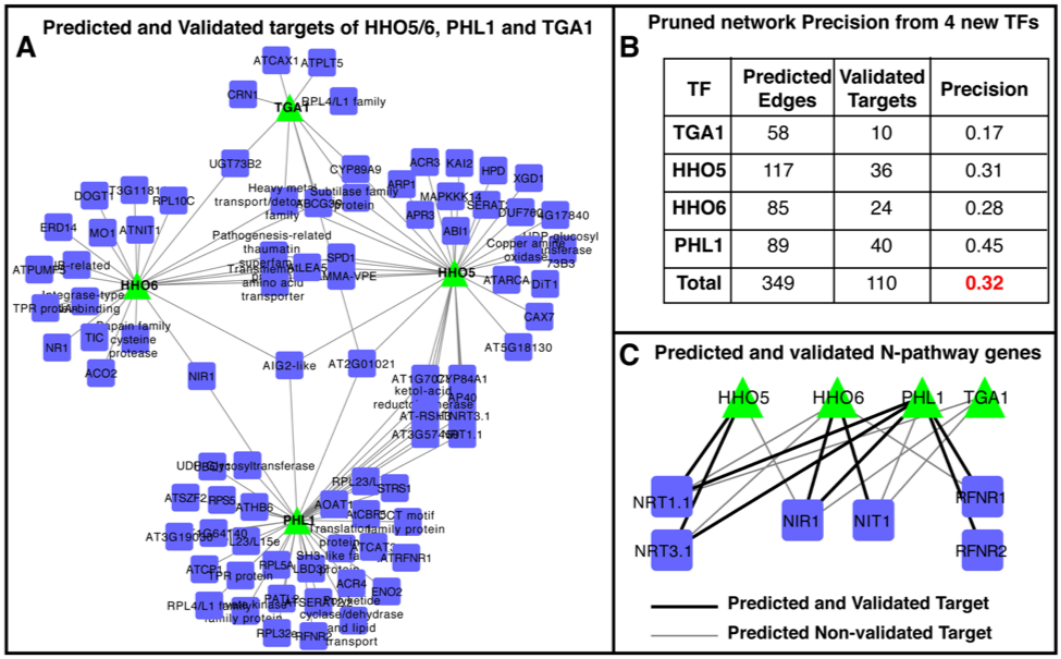
2.3 GRN inference and network pruning

DFG ([58]) was used to infer interactions between 172 TFs and 2,174 NxTime genes in shoots. Experimentally determined TF-target relationships from CRF4, SNZ and CDF1 were used to perform an AUPR analysis and identify a pruning threshold of precision =0.345, which was independently confirmed with validated targets of 4 TFs in the GRN TGA1, HHO5/6, PHL1(Figs. 2.2 and 2.3). Further support for predicted TF-target interactions was obtained from in vitro TF-promoter binding for 40 TFs that have binding data ([73, 44]) (Fig. 2.6). A previously validated machine learning approach that implements Dynamic Factor Graphs [58, 48], was used to derive the TF-target interaction in response to N-treatment. Briefly, the dynamic behavior (i.e., the gene expression values at the nine sampled time points) of the 172 TFs that respond to N-supply in shoots was used to model the behavior of the 2,174 genes responding to N-supply in shoots. Briefly, as we mentioned, Dynamic Factor Graph (DFG) identifies the likely set of TFs driving target gene expression, by learning an f function that explains the target gene expression at each time-point, based on the expression of the TFs at previous time-points [58, 57]. We use the time-series transcriptome data to learn hyper-parameters of the DFG model using a leave-out time-point. Hyper-parameter optimization is the process of choosing a set of hyper-parameters for a good generalization of a learning algorithm [23].

Our dataset contains 10 time points. We train DFG on the first 9 time-points, and tune the hyper-parameters to minimize error on the last time point, and then inferred TF-target edges based on the constructed model. Then we look at the final matrix we obtain using all 10 time points, where the matrix estimates the influence of each TF on every N-responsive gene. This matrix is constructed as a

network with the coefficient of TF influence on a given gene assigned as the edge score for that network edge. This unpruned network contains edges from all TFs to all genes. To increase confidence in the network predictions, we implemented a network pruning approach to used validated TF-targets to identify true positives from false positives, as described in [52]. To implement this Network pruning, we conducted an Area Under the Precision Recall Curve (AUPR) based approach as follows: The true TF-target edges, i.e., experimentally validated edges of 3 TFs (CRF4, SNZ, CDF1. See Figure 2B) were used to calculate and plot the network Precision and Recall (Fig. 2.3C). Briefly, the predicted edges in the DFG-inferred GRN are ranked by their score (i.e., the coefficient of influence of TF on its target gene). The network Precision and Recall are then computed by sliding down the ranked list and labeling each TF-target edge as validated (True positive) or not (False positive). After each step, the Precision (True Positives/ (True + False Positives)) and Recall (True Positives/ (True Positives + False Negatives)) of the network is recalculated (Fig. 2.3A). From these Precision and Recall measures, we determined the minimum edge score that meets a network precision of 0.345 (Fig. 2.3C). This edge score threshold corresponds to 0.95554 in the DFG network. This edge cut-off was chosen to minimize false positives (i.e., higher Precision), while recovering as many true positives as possible. Therefore, all predicted edges with an edge score >=0.95554 (Fig. 2.3C) were retained in the pruned network to generate the pruned TF-target network. This pruned DFG-inferred network represents a highly conservative estimate of the true influence of a TF, and therefore has a low Recall rate compared to the genome-wide targets of each of these TFs. Additional experimental support for TF-> Target interactions was also obtained from independent in vitro TF-binding data [73, 44] (Fig. 2.6). This TF-target DNA binding

dataset included in vitro TF-target binding information for 35 N-responsive TFs with DFG predictions (Fig. 2.6). For each TF with target predictions (DFG) and binding data (DAP-Seq [73, 44]), the two target sets were intersected to identify supported edges i.e., TF is predicted to regulate the target (by DFG) and TF is shown to bind to the target gene promoter (by DAP-Seq) [73, 44] (Fig. 2.6).



TF	Predicted Edges	Validated Targets	Precision
TGA1	58	10	0.17
HHO5	117	36	0.31
HHO6	85	24	0.28
PHL1	89	40	0.45
Total	349	110	0.32

Figure 2.7: Precision scores of TF-target edges in Pruned N-response GRN are independently validated with four additional TFs in the N-response GRN. A. Experimentally validated genome-wide regulated targets of four N-responsive TFs in the GRN (HHO5, HHO6, TGA1 and PHL1) that were not used in the initial network pruning process based on CRF4, SNZ and CDF1 regulated targets (Fig. 2.2), were determined in shoot cells in the TARGET system [13]. The predicted targets of these four new TFs in the Pruned GRN, that were experimentally validated using the TARGET experiments done by our colleagues at NYU Biology, are shown here. B. Precision of the Pruned GRN was re-estimated using the genome-wide taregts of these four new TFs. The overall network precision for this new set of TFs is 0.32 (110 validated out of 349 predicted). Also see Fig. 2.3B. C. Three of these four TFs - HHO5, HHO6 and PHL1 - are predicted and validated to influence six genes in multiple stages of the N-assimilation pathway. TGA1 was predicted to influence NRT1.1, NIR1 and NIT1 genes in the shoot NxTime response, but the TARGET system failed to validate these network predictions (i.e., false positives).

Chapter 3

OutPredict

In this chapter we present a new method called OutPredict. Here, we have tackled the causality problem differently: we first create a model for each gene g which can predict the expression of that gene in out-of-sample time points. If our model is good, then the transcription factors most influential to gene g are likely to be the causative components for g.

The ability to accurately predict the causal relationships from transcription factors to genes would greatly enhance our understanding of transcriptional dynamics. This could lead to applications in which one or more transcription factors could be manipulated to effect a change in genes leading to the enhancement of some desired trait. Here we present a method called OutPredict that constructs a model for each gene based on time series (and other) data and that predicts gene's expression in a previously unseen subsequent time point. The model also infers causal relationships based on the most important transcription factors for each gene model, some of which have been validated from previous physical experiments. The method benefits from known network edges and steady-state data to enhance predictive

accuracy. Our results across B. subtilis, Arabidopsis, E.coli, Drosophila and the DREAM4 simulated in silico dataset show improved predictive accuracy ranging from 40% to 60% over other state-of-the-art methods. We find that gene expression models can benefit from the addition of steady-state data to predict expression values of time series. Finally, we validate, based on limited available data, that the influential edges we infer correspond to known relationships significantly more than expected by chance or by state-of-the-art methods.

3.1 Data

Public datasets vary greatly by organism with respect to experimental design, data density, time series structure and assay technologies. To show its general applicability, we test OutPredict on five different species (Table 1): i) a Bacillus subtilis dataset ii) an Arabidopsis dataset in shoot tissue iii) a Escherichia coli dataset iv) a Drosophila time series dataset, and v) the DREAM4 one-hundred node in silico challenge. When applicable, we denote data as "gold standard" when it is highly curated regulatory or binding data.

B. subtilis: This dataset consists of time series and steady-state data capturing the response of B. subtilis to a variety of stimuli [63]. The gold standard network prior is a curated collection of high confidence edges from high throughput ChIP-seq and transcriptomics assays on SubtiWiki[56] (we used the parsed data set provided in [10]).

Arabidopsis thaliana in shoots [86]: This dataset consists of gene expression level measured from shoots over the 2-hours period during which the plants are treated with nitrogen. As gold standard network data, we used experimentally val-

idated edges from the plant cell-based *TARGET* assay, which was used to identify direct regulated genome-wide targets of N uptake/assimilation regulators[86].

E. coli: This dataset includes the E. coli gene expression values, measured at multiple time points following five distinctive perturbations (i.e., cold, heat, oxidative stress, glucose-lactose shift and stationary phase)[43]. We used as gold standard ancillary data the regulatory interactions aggregated from a variety of experimental and computational methods that has been collected and described in RegulonDB[79]. We retrieved both parsed expression dataset and gold standard data from [38].

Drosophila melanogaster: This dataset consists of gene expression levels covering a 24-hour period; it captures the changes during which the embryogenesis of the fruitfly Drosophila occurs[37]. As gold standard network data, we used the experimentally validated TF-target binding interactions in the DroID database[61]. These interactions come from a combination of ChiP-chip/ChIP-seq, DNAse footprinting, in vivo/vitro reporter assays and EMSA assays across various tissues from 235 publications.

Huynh et al[38] also used this Drosophila data.

DREAM4 synthetic data [34]: This synthetic dataset from the DREAM4 competition consists of 100 genes and 100 TFs (any gene can be a regulator). Because this is synthetic data, the underlying causality network is known.

Dataset	Number of Time-points(Num of Reps)	Steady-State points	Genes	TFs	gold standard edges (TFs)
B. subtilis	7(3),17(1),4(3),10(1),10(1), 11(1),8(1),10(1),11(1) [63]	52(3reps) [63]	4218	239	3144(154) [10]
Arabidopsis[86]	9(3),9(3)	0	2173	162	1731(7)
E. coli	7(3),7(3),7(3),9(3),5(3) [43]	0	2006	163	4899(163) [38]
Drosophila	28(1) [37]	0	1000	14	1660(9) [61]
DREAM4[34]	20 different time series with 11 time-points (1rep)	201(1rep)	100	100	176(41)

Table 3.1: Description of Datasets: the table shows the number of data points in each time series (in parentheses the number of replicates for each data point), available steady-state data, and the number of genes and transcription factors (TFs) under consideration for each species. "Gold standard" data is either well-curated binding data or regulated data or both.

3.2 Methods

Methods

Time series predictions using Random Forests

OutPredict [22] learns a function that maps expression values of all active transcription factors at time t, to the expression value of each target gene (whether a transcription factor or not) at the next time point. Thus, for each gene target, OutPredict learns a many-to-one non-linear model relating transcription factors to that target gene.

The gene function is embodied in a Random Forest, as used previously in Genie3 [39], iRafNet [68], DynGenie3 [38]. When used on a single time series, the Random Forest for each gene is trained on all consecutive pairs of time points except the last time point. For example, if there are seven time points in the time series, then the Random Forest is trained based on the transitions from time point 1 to 2, 2 to 3, 3 to 4, 5 to 6. Time point 7 will be predicted based on the trained function

when applied to the data of time point 6. The net effect is that the testing points are not used in the training in any way because the test set includes only the last time points of each time series.

For a given time series, when multiple time series are available, OutPredict trains the Random Forest on all consecutive pairs of time points (always excluding the last time point) across all time series. Further, OutPredict treats replicates independently, viz. if there are k1 replicates for time point t1 and k2 for subsequent time point t2, then we consider k1 $\times$ k2 combinations in the course of our training. The result of the training is to construct a single function f for each target gene that applies to all time series. To test the quality of function f, we evaluate the mean-squared error (MSE) on the last point of every time series on that target gene.

The Random Forest uses bootstrap aggregation, where each new tree is trained on a sub-sample of the training data points. The Out-of-Bag error for a given training data point is estimated by computing the average difference between the actual value for a given training data point and the predictions based on trees that do not include the training data point in their bootstrap sample. Each tree is built on a bootstrap sample of size approximately 2/3 of the training dataset. Bootstrap sampling is done with replacement, and the remaining 1/3 of the training set is used to compute the out-of-bag score. Thus, the out-of-bag calculation is done on training data only.

All our experiments used random forest ensembles of 500 trees to avoid overfitting. Pruning did not improve the out-of-bag score, so the experiments used the default parameters for pruning of *RandomForestRegressor* in *sklearn* [67].

Incorporation of gold-standard data as priors

OutPredict uses prior data to bias the training of the Random Forest model. Specifically, each decision tree node within a tree of the Random Forest will be biased to include a transcription factor X_1 for the model of gene g in preference to transcription factor X_2 if the prior data indicates a relationship between X_1 and g but none between X_2 and g.

The gold standard for OutPredict is a matrix [Genes $*$ TFs] containing 0s and 1s, which indicates whether we have prior knowledge about the interaction of a transcription factor (TF) and a gene. Hence, if the interaction between a TF and gene g is 1, then there is an inductive or repressive edge; while if it's 0, then there is no known edge.

In order to **compute prior weights** from the gold standard prior knowledge, we assign a value v to all interactions equal to 1 (i.e., the True Positive interactions) and 1/v to the interactions identified by 0 (the set of values tried for v is specified in Supplementary Table S2).

During the tree construction, our Weighted Random Forest, at each node d, selects r candidate features (transcription factors) $X_1, X_2, ..., X_r$ according to the prior weights (Figure 3.1); r is the number of features sampled at each node d, which is set to the square root of the total number of transcription factors.

The r candidate transcription factors are a subset of all transcription factors and are randomly sampled at each tree node, biased based on the weights of the priors, as in iRafNet[68]. In addition, OutPredict calculates the $I(d)$(variance reduction * prior weight) criterion (which is defined below in formula (3) of the *Mathematical Formulation* section) for all the selected subset at each node and branch on the

transcription factor with highest I(d).

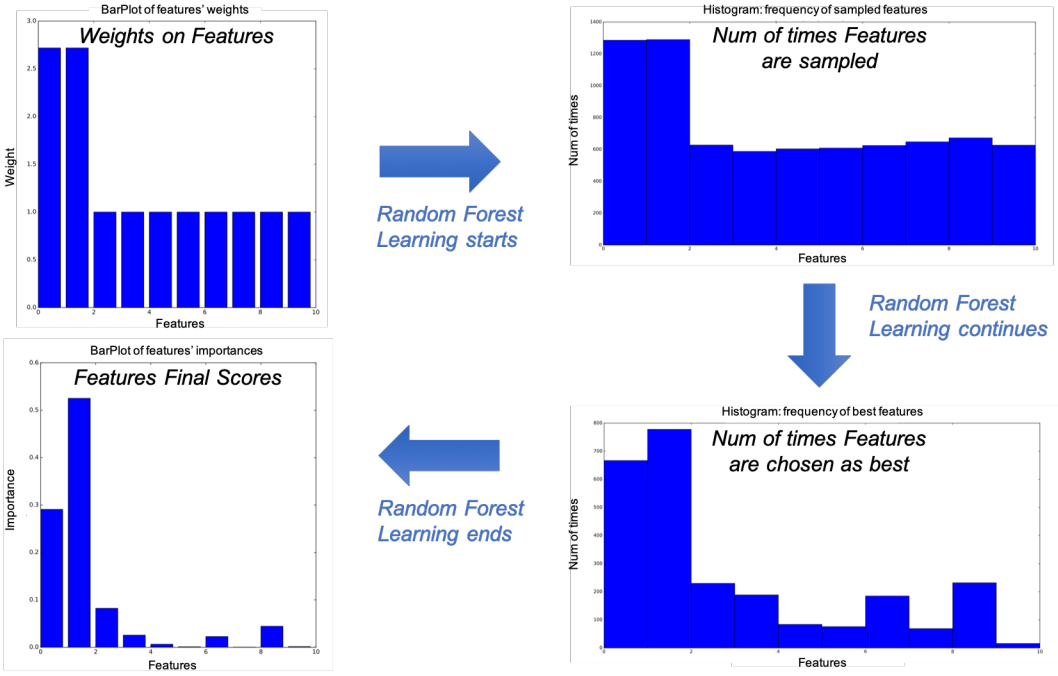


Figure 3.1: Illustration of how priors work: the priors assign initial weights to features (transcription factors) which influence how likely they are to be chosen as splitting elements in the trees of the Random Forest. As learning takes place, these weights can change, finally leading to a model that depends on both the time series data and on other data.

OutPredict incorporates steady-state(SS) data into the same Random Forest model as the time series(TS) data (an "integrated" approach, denoted as the RF_{SS+TS} model). Further, each prior dataset can be evaluated separately depending on how helpful it is to make predictions on time series. By contrast, for example, iRafNet[68], combines all prior datasets and weights them equally at each tree node. An equal weighting strategy may decrease overall performance when, for example, one prior dataset is less informative or is error-rich. As an aside, iRafNet can make out-of-sample predictions but only on steady-state data.

3.3 Mathematical Formulation

Mathematical Formulation

Let X be the expression values of the set of features (in our case, transcription factors), and y_j be a target. We seek a function such that maps X to y_j either in steady-state or for time series. For steady-state data, we use all experimental conditions to infer a function $y_j = fsteady_j(X)$ where X must not include y_j. That is, for each gene y_j, we seek a function from all other genes to y_j. For time series, Outpredict supports two types of models:

1. Time-Step (TS) model:

$$(1) \quad y_j(t_{i+1}) = ftimestep_j(X(t_i)), \forall j$$

2. Ordinary Differential Equation *natural* logarithm (ODE-log) model:

$$(2) \quad \frac{y_j(t_{i+1}) - y_j(t_i)}{ln(t_{i+1} - t_i)} + \alpha y_j(t_i) = fode_j(X(t_i)), \forall j$$

where $X(t_i)$ denotes the expression values of all the transcription factors at time t_i, $y_j(t_{i+1})$ denotes the expression of gene j at t_{i+1}, α is the degradation term. All genes are assumed to have the same α.

OutPredict integrates steady-state(SS) data with Time series(TS) data in a single Random Forest.

We have found that the ODE-log model achieves a better out-of-bag score compared to just using the linear difference $(t_{i+1} - t_i)$ in the denominator. This makes

some intuitive sense because many phenomena in nature show a decay over time. Empirically, for example, the difference in expression value between 5 and 20 is more than 1/3 the difference between 5 and 60 in the Arabidopsis time series. Further, Figure 3.2 illustrates the absolute difference in gene expression decreasing over time for most of the species. During training, one of the Time-Step or

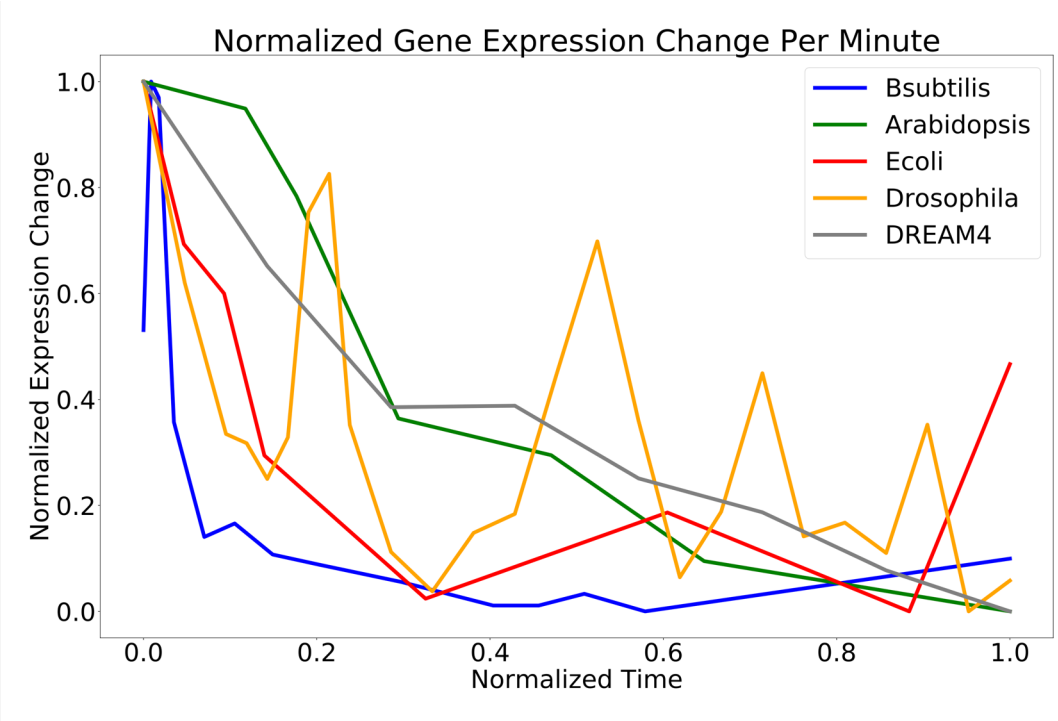


Figure 3.2: Gene Expression Change for all species. Generally, the average absolute difference in expression (across all genes for each species) decreases over time. E. Coli may be an exception because of the short lifespan of bacteria. The Time-Step model worked better for B. subtilis and Drosophila. The Ordinary Differential Equation-log model worked better for Arabidopsis, E. coli and DREAM4 (Table 3.3).

ODE-log models is selected based on the out-of-bag score on the training data. We have found that the relative performances of the two OutPredict techniques Time-Step and ODE-log are very data dependent, with Time-Step performing better than ODE-log on B. subtilis and Drosophila, while the opposite is observed on

Arabidopsis, E.coli and DREAM4 (Table 3.3 shows the best model based on out-of-bag score).

In detail, during training, OutPredict determines (i) which of these two methods (ODE-log or Time-Step) to use, (ii) the prior weights of the TFs, and (iii) the degradation term for the ODE-log model. As far as we know, this is the first time the choice of model and degradation parameter value have been treated as trainable hyper-parameters. We show in Table 3.4 the set of hyper-parameter values tested for the degradation term α and for the prior weights when calculating the out-of-bag score.

Computationally, at a given node d in a tree, OutPredict computes the product of (i) the standard Random Forest importance measure which is defined as the total reduction of the variance of y and (ii) the weight given by the priors. Here is the formula used for the reduction of variance [14], modified by the prior weighting:

$$(3) \quad I(d) = [(S_{num} * var_y(S)) - (S_{l_{num}} * var_y(S_l)) - (S_{r_{num}} * v_y(S_r))] * w_{X_i,y}$$

where d is the current decision node being evaluated, S is the subset of samples that are below decision node d in the tree, S_l and S_r are the subsets of experiments on the left and right branches of decision node d, respectively; var_y is the variance of the target gene in a given subset, and $S_{num}, S_{l_{num}}, S_{r_{num}}$ denote the number of training samples in each subset associated with a specific target gene. Finally, $w_{X_i,y}$ is the prior weight from a given feature X_i to a given target gene y, which causes features with high prior weights to be chosen with higher probability when splitting a tree node during tree construction. Because the model for each target gene is

independent, OutPredict calculates the model for the target genes in parallel. For

Algorithm 1 OutPredict Method

```
1: procedure OutPredict
2:     Split dataset in training and test sets
3:     Test set includes the last time points of all time series
4:     r = sqrt(len(TFs))
5:     if OP-Priors == True then
6:         Compute Prior Weights ▷ (see section on gold-standard data)
7:     For each of the Time-Step and ODE-log models:
8:     Train a Random Forest as follows:
9:     if OP-Priors == True then
10:        Using the training data, do T times
11:        Build a decision tree as follows:
12:        for tree nodes do
13:            Sample r candidates TFs X_1, X_2, ..., X_r according to prior weights
14:            Calculate weighted importance I(d) for these r candidates ▷ (formula 3)
15:            Branch on X_i with highest I_i(d)
16:    else
17:        No priors case: Use training data to build T decision trees for each gene without use of priors.
18:    Return best Time-Step/ODE-log model according to out-of-bag score
19:    Make out-of-sample predictions using test set
20:    Compute importance for each feature
```

the purpose of inferring relative influence of transcription factors on genes and constructing a network of such potential causal edges, let T be the number of trees and D_i be the set of nodes which branch based on transcription factor (feature) X_i, the overall importance score of the feature X_i is:

$$(4) \quad s_i = \frac{1}{T} \sum_{D_i} I(d)$$

Computationally, the importance score s_i of X_i is the sum of the variance improvements I(d) over all nodes d in D_i divided by the number of trees T. The resulting

variable importance value s_i is more robust than the value obtained from any single tree because of the variance reduction resulting from averaging the score over all the trees [14]. High importance scores identify the set of the likely most influential transcription factors for each target gene.

3.4 Results

We measure the prediction performance of our algorithm using the Mean Squared Error(MSE) of the predictions of out-of-sample data. For each species tested, we compare the performance of the different algorithms on time series alone and on time series data with prior information.

As mentioned above, we compared our weighted Random Forest with two related works: i) a Neural Network (NN) with a hidden layer[82] which is an approach developed specifically for time series gene expression prediction. In detail, we perform hyper-parameter optimization for the learning rate of the stochastic gradient descent optimizer, and the dropout rate. Thus, regularization is applied through dropout, which helps reduce overfitting. ii) the Random Forest algorithm DynGenie3 [38], which is an extension of Genie3 [39] that is able to handle both steady-state and time series experiments through the adaptation of the same ordinary differential equation (ODE) formulation as in the Inferelator approach [33]. iRafNet [68], as noted above, does not handle time series data as the main input data.

DynGenie3 was primarily designed for Gene regulatory network inference, but the authors show the performance of DynGenie3 at predicting both time series and steady-state data in the validation sets. Therefore, we evaluate DynGenie3 for

predicting leave-out time series data in order to compare it with OutPredict. As a baseline for all algorithms, we consider the *penultimate value* prediction of the expression of a gene at a given time point to be the same value as the expression of that gene at the immediately previous time point. To evaluate the performance of our forecasting predictions, we compare the predicted expression values to the actual expression values for each gene (Figures 3.3A, 3.4A) and calculate the Mean Squared Error (MSE) across all genes.

Quantitative Results

We show in Figure 3.3B and Figure 3.4B overall bar plots for a *Bacillus subtilis* and *Arabidopsis*. Similar results hold for other species (Figures 3.5, 3.6, 3.7). A table showing which method and data were used for each can be found in Table 3.2. Our basis of comparison is Mean Squared Error, which is a measure of the error in the predictions in which smaller values indicate more accurate predictions. Given a species, the mean squared error (MSE) is calculated as follows: given the prediction and actual value for each replicate of each gene at the last time point, first compute the squared error for each replicate. Second, take the mean to get the mean squared error for that gene. Third, compute the global mean squared error as the mean of the mean squared errors of each gene. Figures 3.3A and 3.4A show qualitatively that the actual values closely track the predicted values. OutPredict outperforms DynGenie3, Neural Nets, and *penultimate value* predictions over all species using these datasets.

In B. subtilis (Fig. 3.3), OutPredict performs 30% better than Penultimate Value ($P < 0.05$, based on a non-parametric paired test), and 50% better than Dynamic Genie3 ($P < 0.05$, based on a non-parametric paired test) (Fig. 3.3B). As Out-

Predict allows the incorporation of priors into the model, such as gold-standard network data, we compared the forecasting performance of OutPredict using time series with the integration of steady-state with OutPredict on time series data with steady-state data and gold-standard regulated edges as priors (Figure 3.8). In these tests, the inclusion of validated gold-standard edges as priors improved predictions compared to excluding priors (Figure 3.8, 11% improvement, $P < 0.05$, non-parametric paired test).

The non-parametric paired test we use throughout this paper compares any two prediction methods M1 and M2 as follows: (i) format the data from the original experiment by a series of rows with one row for each gene containing the gene identifier, the M1 prediction for that gene, the M2 prediction, and the real value (call this series of rows *Orig*); (ii) calculate the figure of merit (for example, the squared error) for each gene and each method (e.g., the square of M1 prediction - real value); (iii) calculate the difference, *Diff*, in the average of the figure of merit (for example, the difference of the mean squared errors) of the M1 values and the M2 values; (iv) Without loss of generality, assume *Diff* is positive; (v) randomization test: for some large number of times N (e.g., N = 10,000), starting each time with *Orig*, for each gene g, swap the M1 and M2 values for gene g with probability 0.5. Now recalculate the overall difference of the figure of merit for M1 and for M2 and see if that difference is greater than *Diff*. If so, that run is considered an *exception*; (vi) The p-value of *Diff* (and therefore of the change in the figure of merit) is the number of exceptions divided by N. When the p-value is small, the observed difference is unlikely to have happened by chance.

We show in Table 3.2 the different models that were compared for the experimental results: each model (built with a given algorithm) is associated with a given

species, a specific main input dataset and a prior dataset. Recall that, in OutPredict, the priors bias the Random Forest by adjusting the weights that determine feature inclusion.

Furthermore, we show the results using the OutPredict (*OP*) technique (either the Time-step or ODE-log) that validation analysis found to be the best model using the out-of-bag score. We found that the weights/importance found in high quality prior data significantly improve predictions in B. subtilis (Fig. 3.3B), though less so in Arabidopsis Shoots (Fig. 3.4B). There is no improvement in E. coli, Drosophila or Dream4 (Figs 3.5, 3.6, 3.7). The precise reasons may vary: gold standard data may contain inaccurate regulatory interactions, may be either incomplete, or may depend on specific experimental conditions. The DREAM4 dataset shows that Priors data contributes to out-of-sample predictions more when there are few time series than when there is abundant time series data (Figures 3.13, 3.14); similarly, the out-of-sample predictions improvement of using time steady-state data, relative to time series data alone, decreases as the number of time series increases (Figures 3.11, 3.12).

Label	Method	Description
OP-Priors	OutPredict-Priors	OutPredict uses (i) Time series(TS) with steady-state(SS) data integrated (TS+SS) in one big Random Forest, and (ii) Gold standard data as priors to bias the integrated Random Forests for time series and steady-state data.
OP-TSonly	OutPredict-TimeSeriesOnly	No Priors: Time series alone; no other data.
DynGenie3	Dynamic Genie3	settings and hyper-parameter optimization as described in [38]
NN	Neural Network	one hidden layer as described in [82]
Pen. Value	Penultimate Value	the second to last time points of each time series is used as the prediction for the last one.

Table 3.2: Legend of Experimental Results.

As a test of the usefulness of OutPredict's importance scores, or measures of influence, for all the TFs on every target gene, we evaluate the *OP-Priors* model importances in Arabidopsis. The dataset consists of 162 TFs on 2173 targets, total-

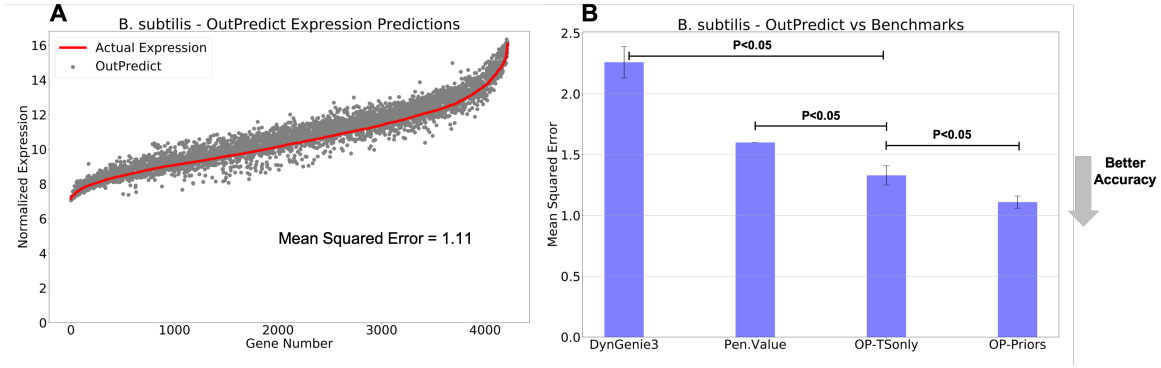


Figure 3.3: Bacillus subtilis. (A) Comparison of predicted gene expression using OutPredict (grey dots) versus actual expression (red line) at the left-out time point. Genes are ordered by increasing actual mean expression value (red line). OutPredict predicts gene expression well at all expression levels. The accuracy of forecasting is measured by calculating the Mean Squared Error (MSE). (B) The vertical axis indicates MSE, where lower bars indicate more accurate predictions. The descriptions of the different models of the x axis can be found in Table 3.2. OutPredict (*OP-Priors*) performs significantly better ($P<0.05$, based on a non-parametric paired test) than *Penultimate Value* (with a 30% relative improvement), DynGenie3 (with a 50% relative improvement) and Neural Network(NN). The MSE for Neural Nets is 3.75 (with standard deviation ≈ 0.3), which is considerably higher than for other methods (Table 3.5); it is not shown here because the MSE is out of scale. Moreover, when priors from both Integrated steady-state data and prior gold standard data, are used with the OutPredict algorithm, there is a significant ($P<0.05$, non-parametric paired test) improvement in predictions relative to OutPredict using only time series data. Specifically, prior gold standard data is significantly helpful, showing a 11% relative improvement (Figure 3.8). Finally, out-of-bag analysis concludes that the Time-step differencing model is better than the ODE-log.

ing 352,026 TF–target edges. To refine these time-based TF–target predictions, we retained the highest-confidence edges, specifically, the top 2% of the edges according to the score, resulting into 7042 edges. We used 1754 validated TF–target edges of 11 TFs physical experiments from [74],[11],[12],[49],[30],[53],[35],[17] (the data for the 11 TFs are described in Table 3.6), which is a disjoint dataset from the one used for the priors. This analysis establishes the precision (i.e., the proportion of

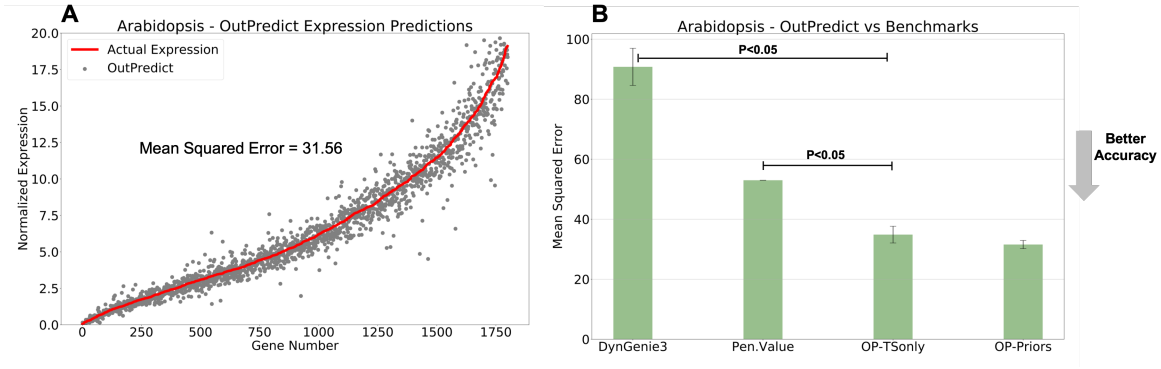


Figure 3.4: Arabidopsis in Shoot Tissue (time series only dataset) (A) Predicted gene expression using OutPredict (grey dots) compared to actual expression (red line) at the left-out time point. (B) Comparison of time series forecasting: the accuracy of forecasting, measured by Mean Squared Error, has higher values in this case than for other species, because the data is RNAseq and read counts have a broad dynamic range. Table 3.2 describes which method and data were used for each model in the x axis. OutPredict (*OP*) performs 34.2% better than *Penultimate Value* ($P < 0.05$, non-parametric paired test), and 61.5% better than Dynamic Genie3 ($P < 0.05$, non-parametric paired test). The incorporation of priors from *TARGET* (*OP-Priors*) improves the performance of OutPredict compared to the time series alone (9% improvement with $P = 0.12$, non-parametric paired test). The ODE-log model is better than Time-Step based on the out-of-bag score. The Neural Network model doesn't converge because the dataset is small.

Dataset	Best OutPredict Model
B. subtilis	Time-Step (7% better than ODE-log)
Arabidopsis	ODE-log (22% better than Time-Step)
E. coli	ODE-log (15% better than Time-Step)
Drosophila	Time-Step (17% better than ODE-log)
DREAM4	ODE-log (5% better than Time-Step)

Table 3.3: Time-Step(TS) vs ODE-log model. For a given organism the table shows the best model based on out-of-bag score. The relative performance of the two OutPredict techniques Time-Step and ODE-log are very data dependent, with Time-Step performing better than ODE-log on B. subtilis and Drosophila, while the opposite is observed on Arabidopsis, E.coli and DREAM4. We determine this on the training data and then apply whichever method is better on the test data.

predicted TF-target edges that are validated) and recall (i.e., the proportion of validated TF-target edges that are predicted) of the OutPredict top 2% edges for the

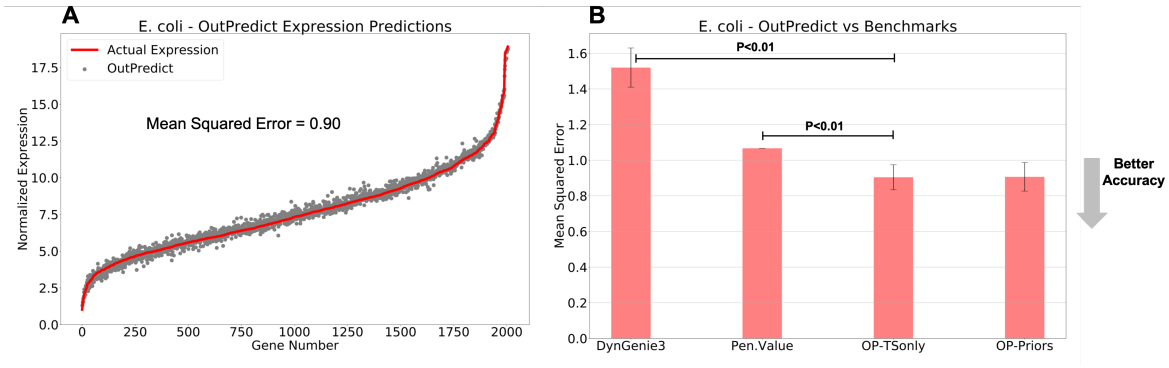


Figure 3.5: **Escherichia coli: Time series forecasting.** This is a time series only dataset consisting of 15 time series. (A) Comparison of predicted gene expression using OutPredict (grey dots) vs. actual expression (red line) at the left-out time point. The accuracy of forecasting is measured by calculating the Mean Squared Error. (B) OutPredict (*OP* and *OP-Priors*) improves ($P < 0.01$, based on a non-parametric paired test) the quality of forecasting compared to *Penultimate Value* (15% improvement) and Dynamic Genie3 (40.5% improvement). For this data, there is no improvement using priors from gold-standard edges compared with time series data by itself.

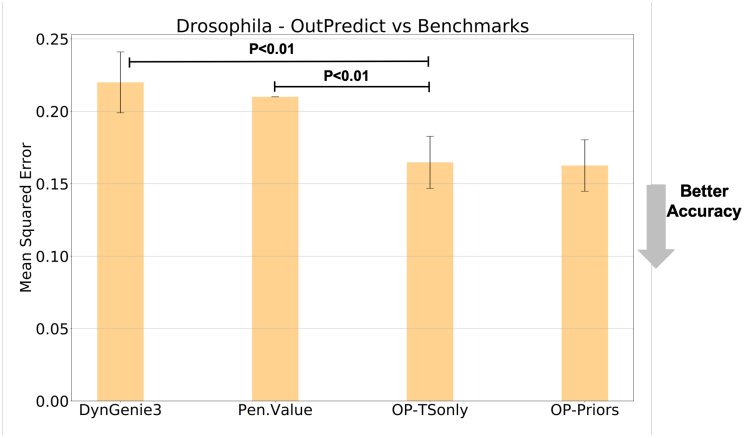


Figure 3.6: **Drosophila: Time series forecasting.** This is a time series only dataset consisting of one time series of 28 time-points. OutPredict (*OP* and *OP-Priors*) performs better ($P < 0.01$, non-parametric paired test) than benchmark approaches including *Penultimate Value* and Dynamic Genie3 (23% and 26.1% improvement, respectively). The incorporation of priors from the gold-standard network does not improve forecasting compared to time series alone.

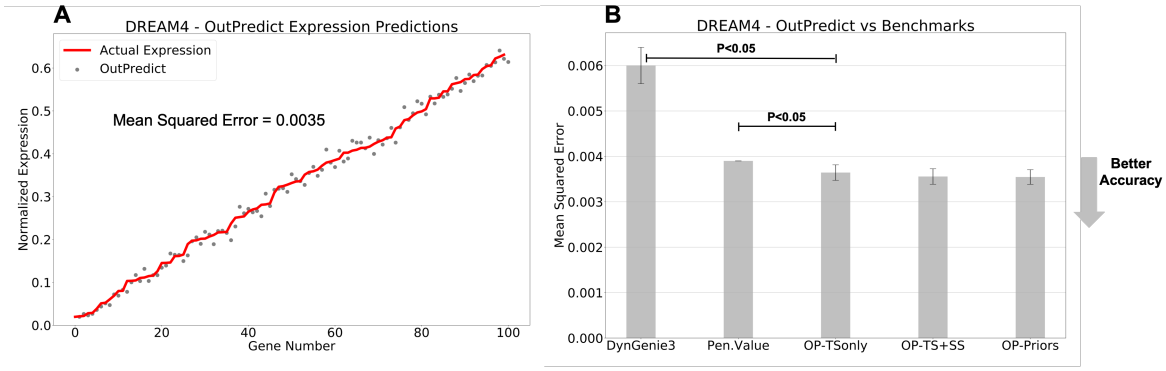


Figure 3.7: **DREAM4: Time series forecasting.** This is a synthetic dataset. (A) Comparison of predicted gene expression using OutPredict (grey dots) vs. actual expression (red line) at the left-out time point. (B) OutPredict (*OP-TSonly*, *OP-TS+SS* and *OP-Priors*) outperforms ($P < 0.05$, non-parametric paired test) *Penultimate Value* and Dynamic Genie3 with 10% and 40.1% relative improvement, respectively. The incorporation of priors together with the integration of steady-state data does not improve forecasting compared to time series alone.

Hyper-parameter	Set of values tested
alpha (α)	[1, $2e{-}1$, $1e{-}1$, $4e{-}2$, $2e{-}2$, $1e{-}2$]
prior weights (True Positive)	[2,exp(1), 5, 8, 15]

Table 3.4: Hyper-parameters: Set of values tested for the degradation term alpha (α) and for the prior weights when calculating the out-of-bag score. As explained in the body of the paper, when OP-Priors is set to $True$ and gold standard data is given as priors, OutPredict transforms the gold standard prior knowledge to prior weight, by assigning a value v (chosen from the set of prior weights in the table) to all interactions where there is an edge in the prior data and 1/v to the interactions where the existence of an edge is unknown.

validated 11 TFs. The results showed that precision and recall for the TF–target predictions in the top 2% edges were 0.246 (76/309) and 0.043 (76/1754), respectively. Both were significantly greater than the mean for 1000 random samples of 309 edges of these 11 TFs (random precision mean ≈ 0.161 and random recall mean ≈ 0.028) (Table 3.7). Moreover, the precision of OP-Priors for the top 2% outperforms OP-TSonly (precision=0.226) and DynGenie3 (precision=0.158). We further compared the performance of the OP-Priors model importances with OP-

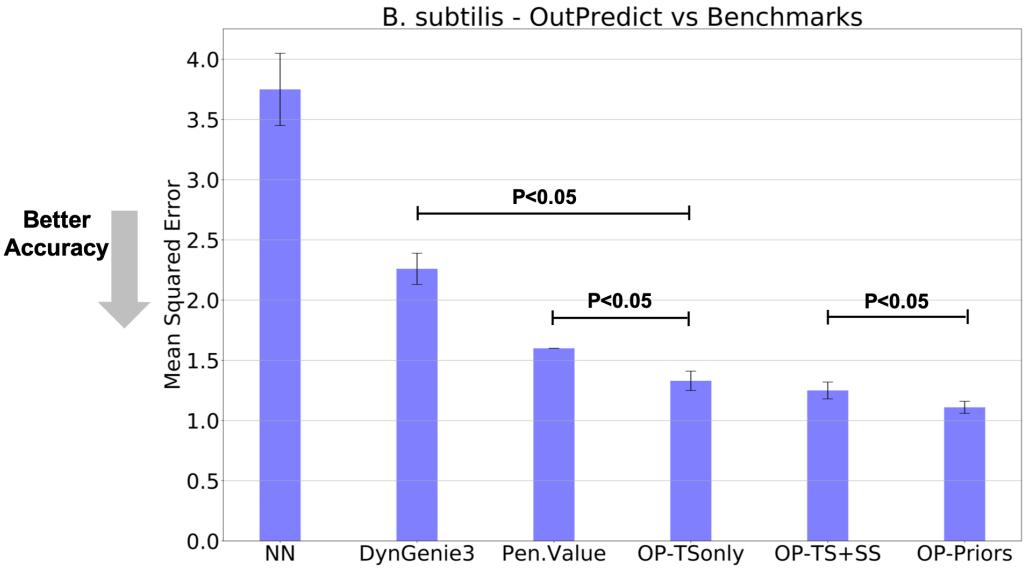


Figure 3.8: - Bacillus Subtilis - Full Comparison of time series forecasting: Neural Network from [Smith et al 2010] (NN) vs. Dynamic Genie3 (DynGenie3) vs. Penultimate Value (Pen.Value) vs. OutPredict (*OP-TSonly*, *OP-TS+SS* and *OP-Priors*). The use of steady-state data (*OP-TS+SS*) leads to a 6% significant improvement ($P < 0.05$, non-parametric paired test) relative to time series data alone (*OP-TSonly*). *OP-Priors* uses gold standard data (in addition to time series (TS) and steady-state (SS) integrated in a single random forest), which is helpful compared to the model *OP-TS+SS* showing an 11% relative improvement ($P<0.05$, non-parametric paired test).

Dataset	Neural Netwok MSE (StdDev)	OutPredict Time-Series-only MSE (StdDev)
B. subtilis	3.75 (0.3)	1.33 (0.08)
E. coli	3.33 (0.27)	0.9044 (0.07)
DREAM4	0.0095 (0.0008)	0.0036(0.00017)

Table 3.5: Neural Network (NN) with one hidden layer [Smith et al 2010] vs. OutPredict Time-Series-only (OP-TSonly). NN from [Smith et al 2010] is able to learn using time series only datasets. The table shows that the mean squared error (MSE) for NN is significantly higher than for OutPredict since there is a relatively small amount of data. Neural Networks work best with much larger datasets. NN doesn't converge for Arabidopsis and Drosophila because the datasets are too small.

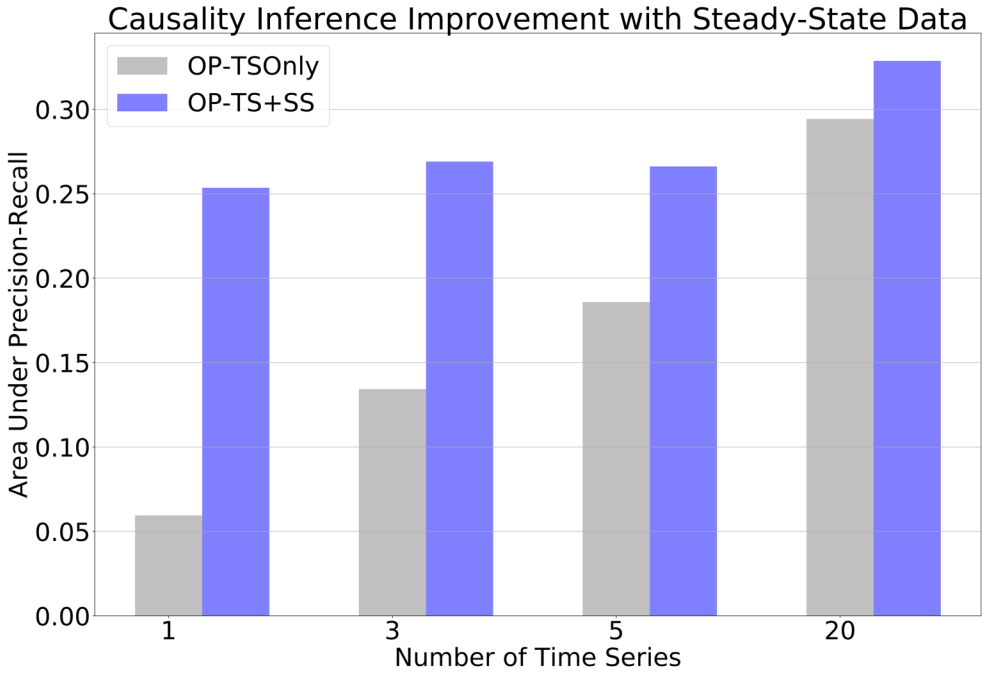


Figure 3.9: DREAM4 - Causality Inference Improvement with Steady-State data. The DREAM4 dataset shows that steady-state data contributes to the inference of causality more when there are few time series than when there is abundant time series data. (A) We show the comparison of Area under Precision-Recall (AUPR) with and without steady-state data in cases of different numbers of time series. The y-axis represent the AUPR average of three different random sets of time series of size $1, 3, 5$ respectively; $x = 20$ represents the case of taking all 20 time series in the DREAM4 dataset.

TSonly and DynGenie3, and computed the Area under Precision-Recall (AUPR) using the 1754 validated TF–target edges of 11 TFs physical experiments in Arabidopsis. The AUPR of Outpredict with Priors (OP-Priors) is 15% better than random (p-value < 0.01, non-parametric paired test), for Outpredict without Priors (OP-TSonly) AUPR is 7.5% better than random (p-value < 0.01, non-parametric paired test), while DynGenie3 is no better than random (Figure 3.15). Additionally, we show that similar results (Figure 3.16) hold for the DREAM4 synthetic dataset (where causal edges are known). This shows the promise of using predic-

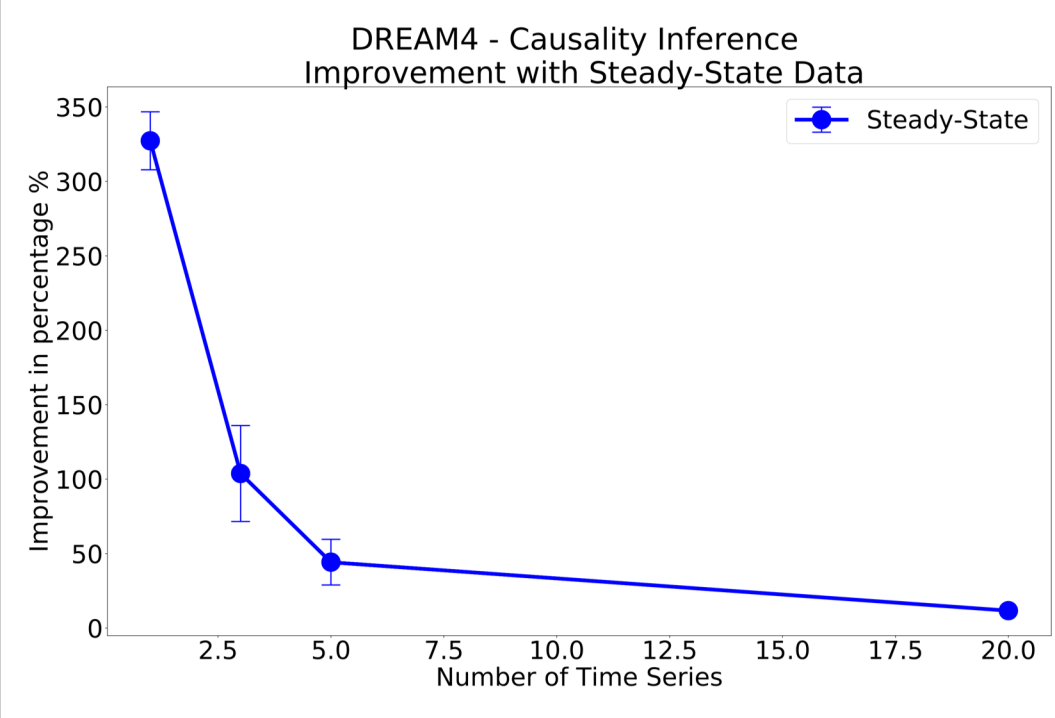


Figure 3.10: DREAM4 - Causality Inference Improvement with Steady-State data. The DREAM4 dataset shows that steady-state data contributes to the inference of causality more when there are few time series than when there is abundant time series data. The AUPR improvement of using time steady-state data, relative to time series data alone, decreases as the number of time series increases.

tion to infer influence and suggests that good out-of-sample prediction leads to good causality models.

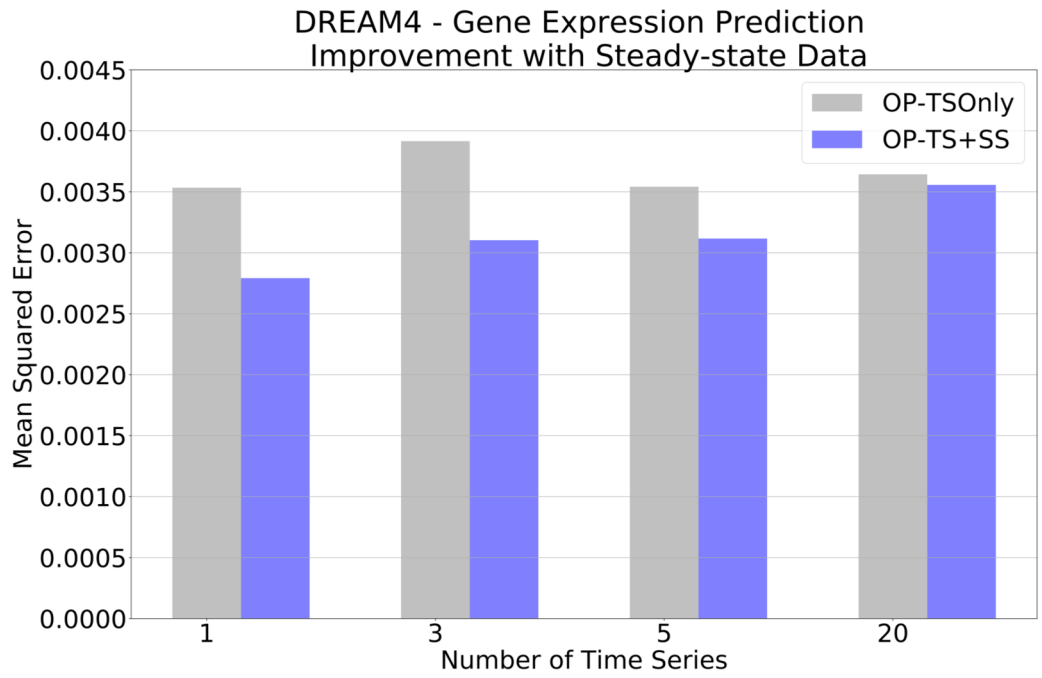


Figure 3.11: DREAM4 - Gene Expression Prediction Improvement with Steady-State data. The DREAM4 dataset shows that steady-state data contributes to out-of-sample prediction more when there are few time series than when there are many. (A) We show the comparison of time series forecasting with and without steady-state data for different numbers of time series. The y-axis represent the MSE (mean squared error) average for three different random sets of time series of sizes $1, 3, 5$ respectively; $x = 20$ represents the use of all 20 time series in the DREAM4 dataset.

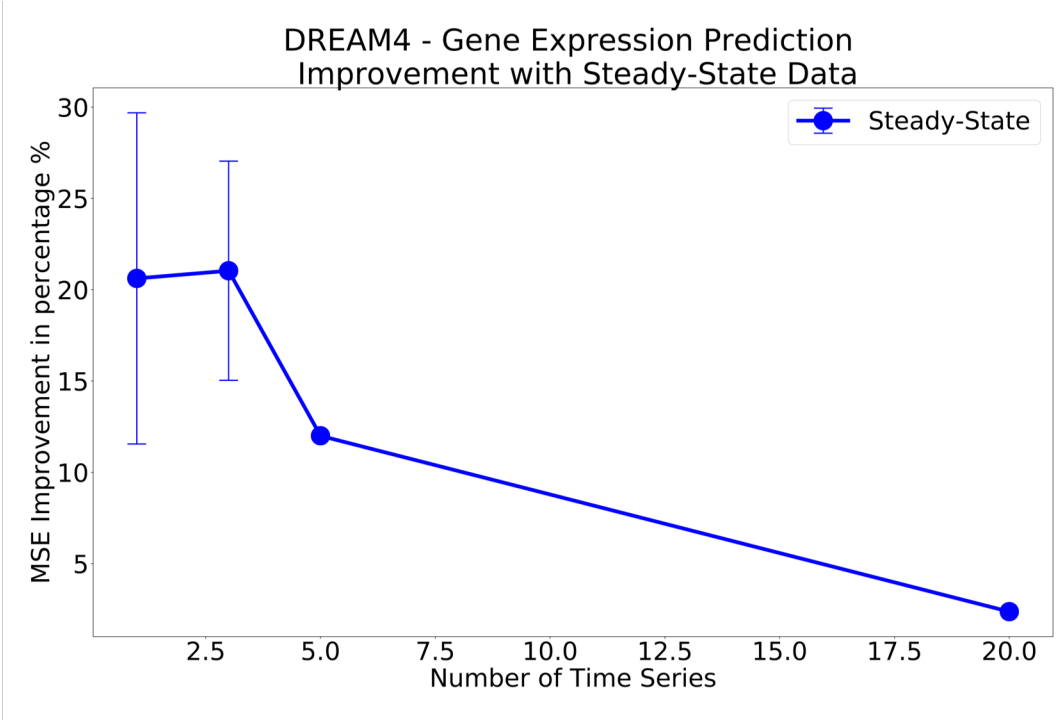


Figure 3.12: DREAM4 - Gene Expression Prediction Improvement with Steady-State data. The DREAM4 dataset shows that steady-state data contributes to out-of-sample prediction more when there are few time series than when there are many. The out-of-sample predictions improvement of using time plus steady-state data, relative to time series data alone, decreases as the number of time series increases.

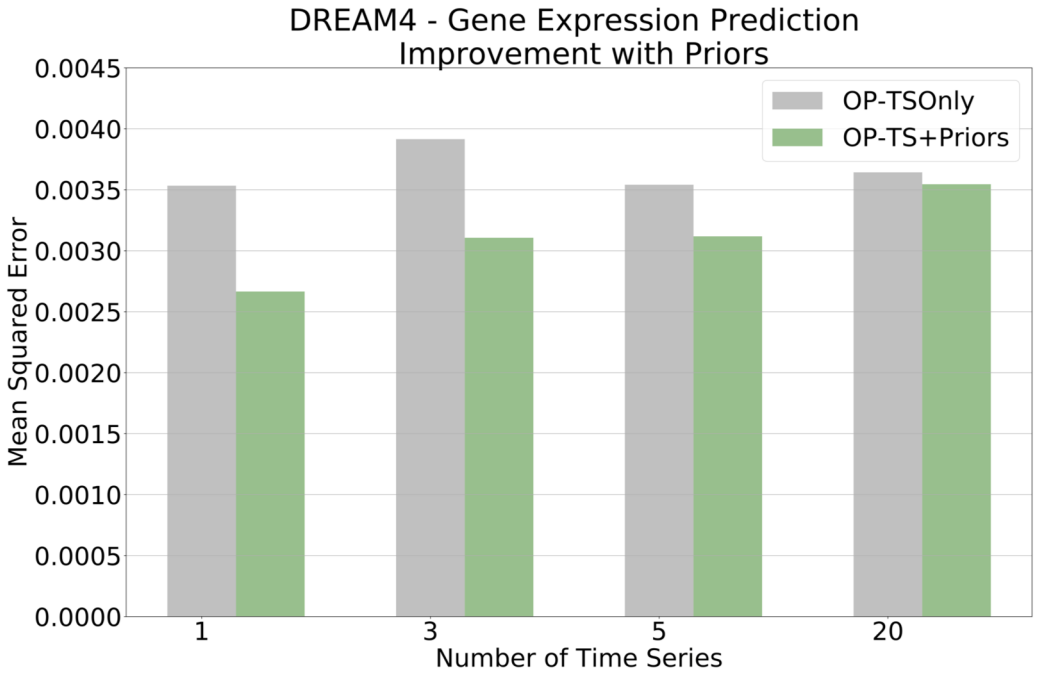


Figure 3.13: DREAM4 - Gene Expression Prediction Improvement with Priors. The DREAM4 dataset shows that Priors data contributes to out-of-sample predictions more when there are few time series than when there are many. (A) We show the comparison of time series forecasting with and without gold standard data for different numbers of time series. The y-axis represent the MSE (mean squared error) average for three different random sets of time series of size $1, 3, 5$ respectively; $x = 20$ represents the use of all 20 time series in the DREAM4 dataset.

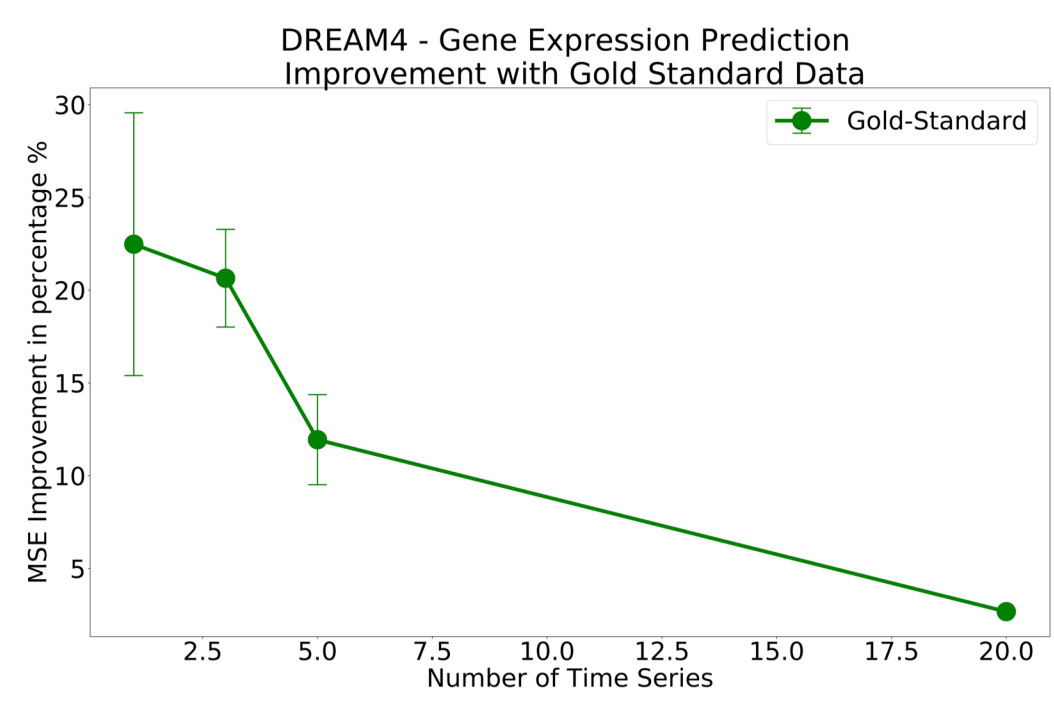


Figure 3.14: DREAM4 - Gene Expression Prediction Improvement with Priors. The DREAM4 dataset shows that Priors data contributes to out-of-sample predictions more when there are few time series than when there are many. Therefore, when the gold standard as priors is used in addition to time series data, the out-of-sample prediction improvement decreases as the number of time series increases.

Transcription Factor	Technology
CGA1/GNL(AT4G26150)	Microarray-Agilent
GATA17(AT3G16870)	Microarray-Agilent
GATA2(AT2G45050)	Microarray-ATH1
LBD38(AT3G49940)	Microarray-ATH1
LBD37(AT5G67420)	Microarray-ATH1
PHR1(AT4G28610)	Microarray-ATH1
NLP7(AT4G24020)	Microarray-CATMA
HBI1(AT2G18300)	RNA-seq
CRF4(AT4G27950)	RNA-seq
GNC(AT5G56860)	Microarray-Agilent combined with RNA-seq
SVP(AT2G22540)	RNA-seq

Table 3.6: The Transcription Factor (TF) experiments used for the validation of OutPredict's Arabidopsis Model importance output. Regarding the Microarray experiments, the genes not on chip were filtered from the predictions according to the microarray type. The microarray elements for the different types were retrieved from the following public repository: CATMA in arabidopsis.org ; ATH1 in arabidopsis.org ; Agilent in arabidopsis.org.

Validated TF-target measures	OP-Priors
Precision/Recall TF-target	0.246/0.043
Random Precision/Recall average	0.161/0.028
Validated Precision/Recall p-value	<0.01/<0.01

Table 3.7: TF-target validation for *OP-Priors* Arabidopsis Model. The important edges predicted by the model had a precision and recall of over 23% and 4%, respectively. Whereas a random selection of the same number of edges had a precision and recall of 16% and under 3% (respectively). The differences for both are statistically significant.

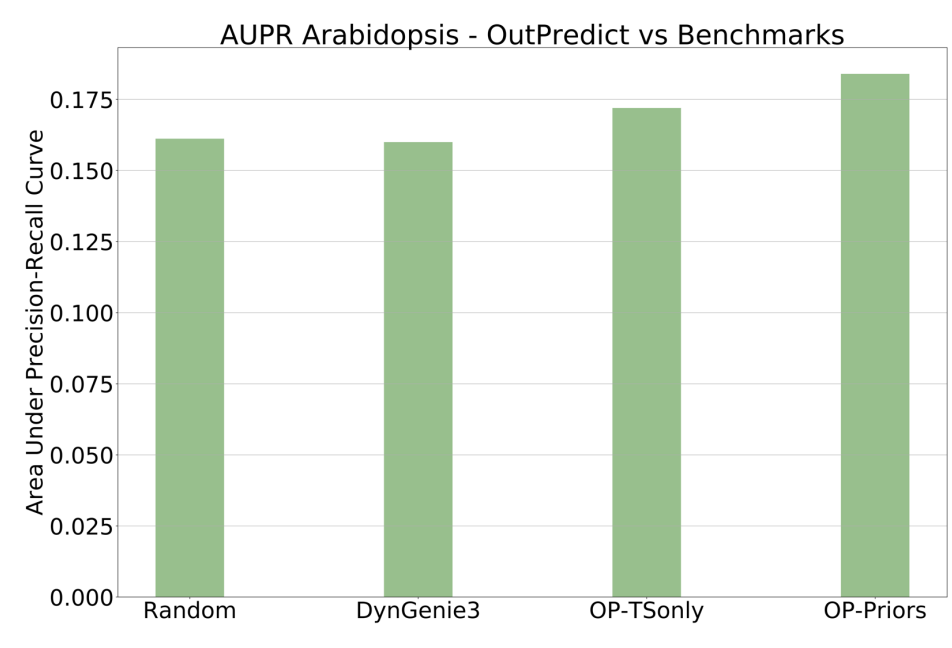


Figure 3.15: Inference of Causality. The area under the precision recall curve (AUPR) of Outpredict with Priors (OP-Priors) is 15% better than random (p-value < 0.01, based on a non-parametric paired test); AUPR of Outpredict without Priors (OP-TSonly) is 7.5% better than random (p-value < 0.01, non-parametric paired test); DynGenie3 same as random.

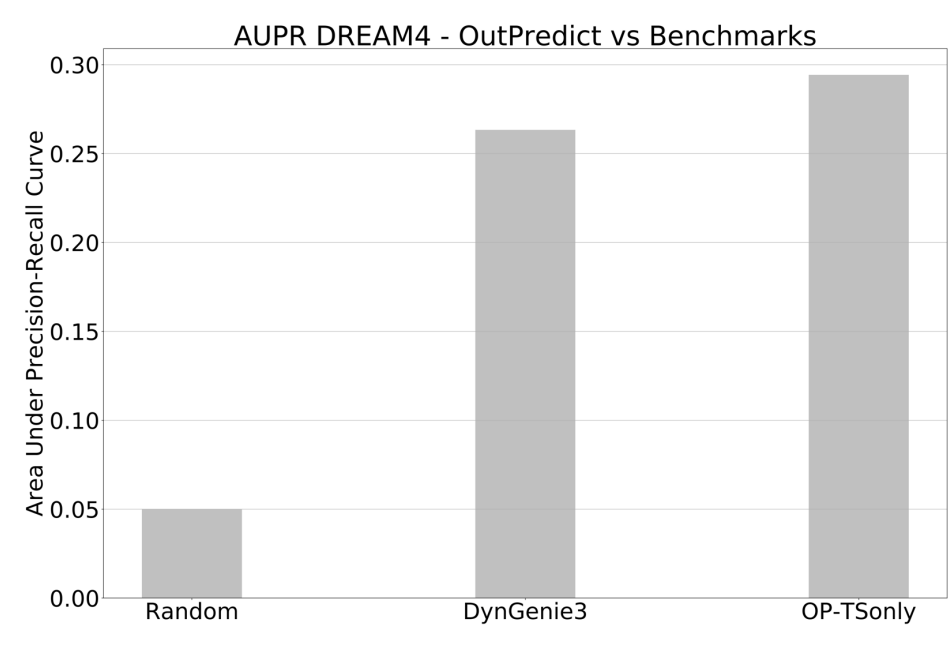


Figure 3.16: AUPR DREAM4 - OutPredict vs. Benchmarks for the inference of causal edges. As for the Arabidopsis dataset (Figure 4 of the main paper), here we show the AUPR (Area Under the Precision-Recall curve) for predicting causal edges in the ideal case of DREAM4 where the true gold standard is known. Outpredict without Priors (OP-TSonly) is clearly better than random (p-value $<$ 0.01, non-parametric paired test) in terms of Area under Precision-Recall. Further, AUPR of OP-TSonly is 12% better than AUPR of DynGenie3 on time series data (p-value $<$ 0.01, non-parametric paired test). This suggests that good out-of-sample prediction leads to good causality models.

OutPredict vs. Dynamic Factor Graph (DFG)

In Chapter 2 and Appendix 6, A.1.8 we discussed our studies where we designed and implemented a computational pipeline that includes the Dynamic Factor Graphs (DFG)([58]) algorithm, which is a State-space model (SSM) algorithm that models the dynamics of a sequence of data by encoding the joint likelihood of observed and hidden variables.

DFG uses an Expectation-Maximization (EM) algorithm which consists of a two-

steps iterative procedure: the step which infers the latent variables $z(t)$ (i.e. the inference step) and the step to learn the linear f function (i.e. the learning step). In the inference step the model infers the latent variable $z(t)$ which represents the denoised version of the gene expression data under the assumption that mRNA data $y(t)$ are noisy observations [57].

In this subsection we compare DFG with OutPredict; as shown in Chapter 2, DFG learns the rate of change of gene expression, therefore we use as performance metric the prediction of the direction of gene change (up-regulation or down-regulation) on future data points. For this purpose we have generate a synthetic $DREAM4$ time-series dataset with 100 genes [80] in order to compare the two algorithms. For each gene both algorithms are trained on all consecutive pairs of time points except the last time point, hence the test set includes only the last time points of each time series.

We use the prediction of the direction of change to gene expression to count the number of times the sign of the change between the penultimate and last time point for all time series is correct, which we call signs on leave-1 test dataset.

The leave-out-last predictive performance on the last time points for DFG was worse than OutPredict, with 56% and 67.5% of correct signs on the test set respectively (Figure 3.17).

We further compared the performance of the OP-TSonly model importances with DFG, and computed the Area under Precision-Recall (AUPR). This is possible because the causal edges are known for the DREAM4 synthetic dataset. Figure 3.18 shows that the AUPR (area under the precision-recall curve) of Outpredict without Priors (OP-TSonly) is significantly better than random and DFG (p-value < 0.01, non-parametric paired test). For the sake of fairness, we show the AUPR

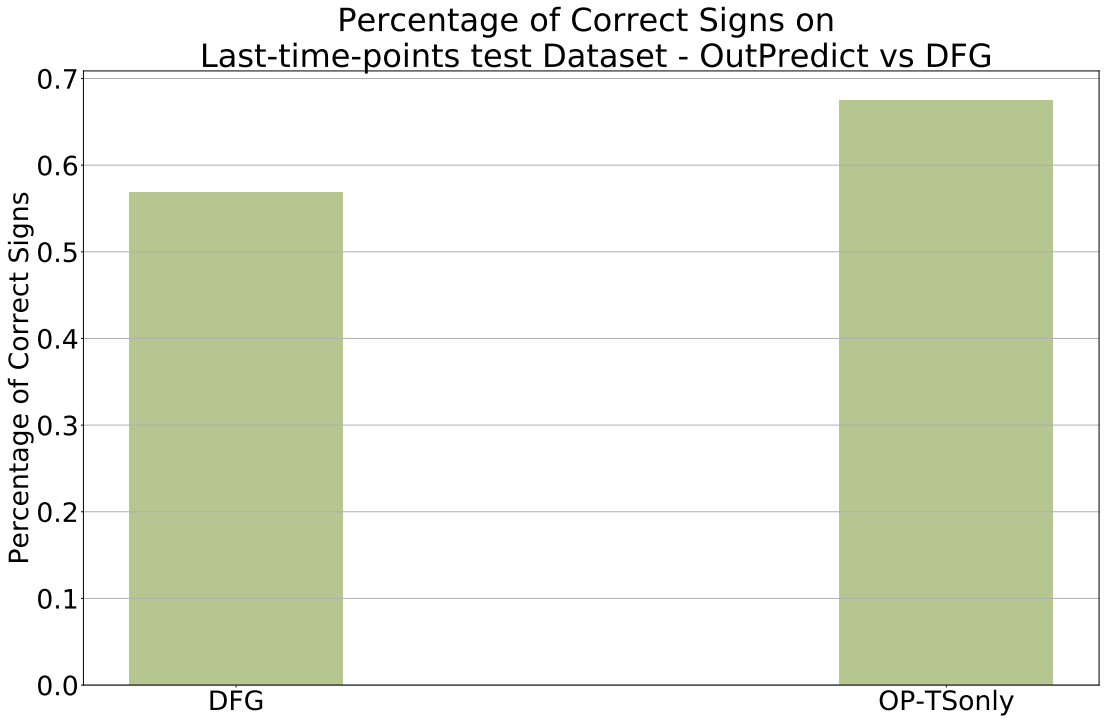


Figure 3.17: Percentage of Correct Signs on last-time-points dataset - DREAM4 - OutPredict vs. DFG. We make predictions about whether gene expression levels would be increased (positive sign) or decreased (negative sign) at the last time-point compared with penultimate (for all time-series). Outpredict (OP-TSonly) is better than DFG (p-value < 0.01, non-parametric paired test) in terms of Percentage of Correct Signs on the last-time-points test dataset for DREAM4 time series.

for DynGenie3 as well. The fact that better prediction corresponds to better feature importances estimation suggests that successful out-of-sample prediction results in strong models of causality.

Conclusion

There are four reasons for the relative success of OutPredict compared to other methods: (i) the use of Random Forests which provides a non-linear model (in

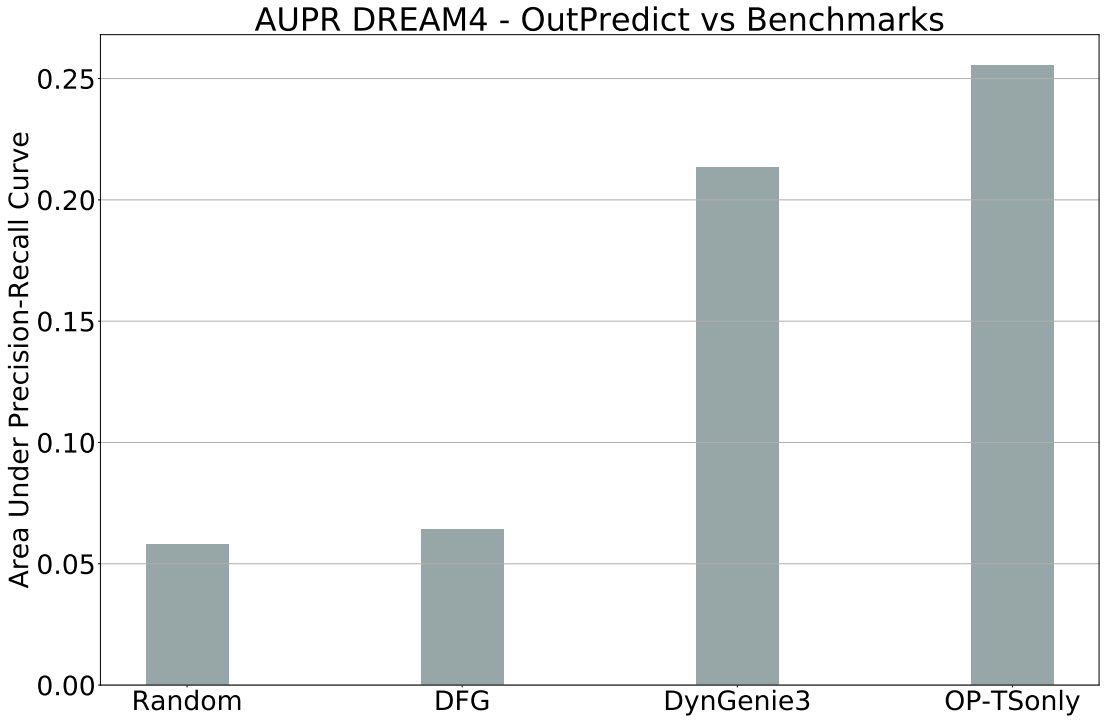


Figure 3.18: AUPR DREAM4 - OutPredict vs. DFG for the inference of causal edges. Here we show the AUPR (Area Under the Precision-Recall curve) for the prediction of causal edges for DREAM4 where the true gold standard is known. Outpredict (OP-TSonly) performs far better than DFG (p-value < 0.01, non-parametric paired test) in terms of Area under Precision-Recall, i.e. 0.26 and 0.064 AUPR values respectively. Further, AUPR of OP-TSonly better than AUPR of DynGenie3 on time series data (p-value < 0.01, non-parametric paired test) as well.

contrast to regression models) that requires little data (in contrast to neural net approaches), (ii) the incorporation of prior information such as gold standard network data (in contrast to DynGenie3), (iii) the adjustment of weights of predictors (in contrast to all other time series based methods), and iv) the selection during training of the optimal technique between the Time-Step and our *ODE-log* model, which includes a degradation term that is also tuned (in contrast to all other methods).

In summary, OutPredict achieves high prediction accuracy and significantly outperforms baseline and state-of-the-art methods on data sets from four different species and the in silico DREAM data as measured by mean squared error. Further, as a proof of concept, we have seen that the high importance edges correspond to individually validated regulation events much greater than by chance in both Arabidopsis and DREAM. The code is open source and is available at the site github.com/jacirrone (*DOI-10.5281/zenodo.3611488*).

Chapter 4

Learning with Steady-State Data Alone

4.1 Single-cell analysis reveals a framework for understanding cell behavior from its birth to terminal differentiation

We apply OutPredict to a single-cells dataset where each single-cell expression profile is processed as a steady-state condition. While a host of studies published in 2019 have largely been descriptive, in this chapter we briefly describe work where we use machine learning modeling together with genetics, and molecular analysis to uncover functional circuitry in a case study focused on the Arabidopsis phloem. Conceptually, this work shows how the morphogen-like gradients that have been implicated in the maturation of the meristem connect to the specific mechanisms that allow individual cell types to time the steps of their differentiation. The

experimental part of this work done by our biologist colleagues at University of Cambridge and University of Helsinki shows these connections directly with genetic perturbations and in-vivo chromatin binding assays. Furthermore, our data and computational models allowed us to form a unique view of cell type differentiation, leading to a model of disconnected developmental changes that are mediated by a so-called *seesaw* model in which early and late stage regulators antagonistically regulate each other's targets.

Overall, these results provide new general principles of how the plant meristem functions, with implications in efforts to improve plant growth.

4.1.1 Single-cell expression profiles as steady-state conditions

Plant roots consist of several concentric layers of functionally distinct cell files, which initially bifurcate and establish distinct identities around the quiescent center and its surrounding stem cells. Cells within each file mature through the distinct zones of cell proliferation and differentiation. In Arabidopsis the development of the protophloem sieve elements (PSEs) involves a transient period of cell proliferation, during which, in addition to amplification of cells within the file, two lineage-bifurcating events take place. Soon after the cell proliferation ceases, cells of the PSE lineage initiate a differentiation process which culminates in enucleation, an irreversible process that gives rise to the mature conductive cells. Because of specific modulation of the graded distribution of the key phytohormonal cue auxin, the differentiation of PSEs occurs faster than that of the other cell files. Therefore, PSE development offers a tractable scheme to understand how the two processes

of cell specialization and maturation interact.

This work concerns data produced by our experimental colleagues at the University of Cambridge and University of Helsinki. The data consists of 758 single-cell PSE transcriptomes. The 758 PSE cells were placed on a linear developmental trajectory using unsupervised tSNE clustering in order to project them into a pseudo-temporal order. This way a pseudotime value is assigned to each cell in order to obtain pseudo-time order of the 758 protophloem cells.

To model gene regulatory connections, we first selected the 15% most variable genes among the 758 cells using the genevarfilter function in Matlab (*Percentile*, 85), leaving 4,924 genes for model inference. We modeled gene regulation using *OutPredict* on the 758 single-cell expression profiles and the 4,924 highly variable genes, which included 208 transcription factors (TFs). In general, the Random Forest model allows for non-linear dependencies of target genes on causal transcription factors. Each single-cell expression profile is treated as a steady-state condition, allowing the model to learn a function that maps expression values of TFs to the expression value of each target gene. In *OutPredict*, the TF-to-target association is described with a *score* that reflects the contribution of the TF to the expression of its target according to the model. To address drop-out effects and other noise in single-cell data in the pre-processing stage, we merged the expression of consecutive cells to generate pseudo-cells using the following procedure to optimize the "bin size" (number of consecutive merged cells): we subdivided the 758 single-cell expression profiles into varying bin sizes, taking the median of the expression value of each gene in each bin or pseudo-cell as the value of that pseudo-cell. The Random Forest approach uses bootstrap aggregation, where each

new tree is trained on a bootstrap sample of the training data. The remaining out-of-bag error is estimated as the average error for each training data point p_i evaluated on predictions from trees that do not include p_i in their corresponding bootstrap sample. For the dataset, the optimal bin size that minimized the out-of-bag error was 12 cells, providing our steady-state inference model a total of 64 pseudo-cells.

Finally, *OutPredict* ranks TFs based on their influence (score) on target gene expression, generating a predicted gene regulatory network (GRN) based on TF causality. To refine these TF-target predictions, we retained the top-10 highest scoring transcription factors for each gene target, resulting in 49,240 (TF-to-target) edges. TFs were then ranked by their number of targets to derive the ranked list of the most important TFs.

In summary, our *OutPredict* gene regulation model is based on the pseudotime-ordered 64 pseudo-single-cell profiles, 208 TFs, and 4924 highly variable genes. We extrapolate four forms of significant and insightful validation:

- Among the 208 TFs in this dataset, the majority of known PSE transcription factors (such as APL, NAC45 and NAC86) were among the top 20 regulators.
- We validated the model by comparing predicted targets with genes induced by in vivo ectopic expression of the same TFs, confirming a significant overlap of targets in 3 out of 5 cases.
- Among the top 20 regulators we also identified four related genes that encode early SE abundant PEAR transcription factors (Figure 4.1). The simulta-

neous loss of six PEAR genes was recently shown to result in defects in PSE differentiation [59].

- TF targets were classified into positive vs. repressive downstream sets in both the *OutPredict* model and experimental over-expression data as follows: Pearson correlation between each TF and individual targets was used to determine regulatory effects (negative correlation, $r<0$, was classified as repressive regulation, and $r>0$ was classified as positive regulation). We evaluated significant overlap between all pairwise positive and negative regulatory sets for each transcription factor (seesaw model) using the Fisher Exact test in Matlab. For the heatmaps in Figure 4.2, the binary output of the Fisher Exact test ($p<0.05=1$, $p>0.05=0$) was multiplied by the fraction of overlap between the two TF target sets.

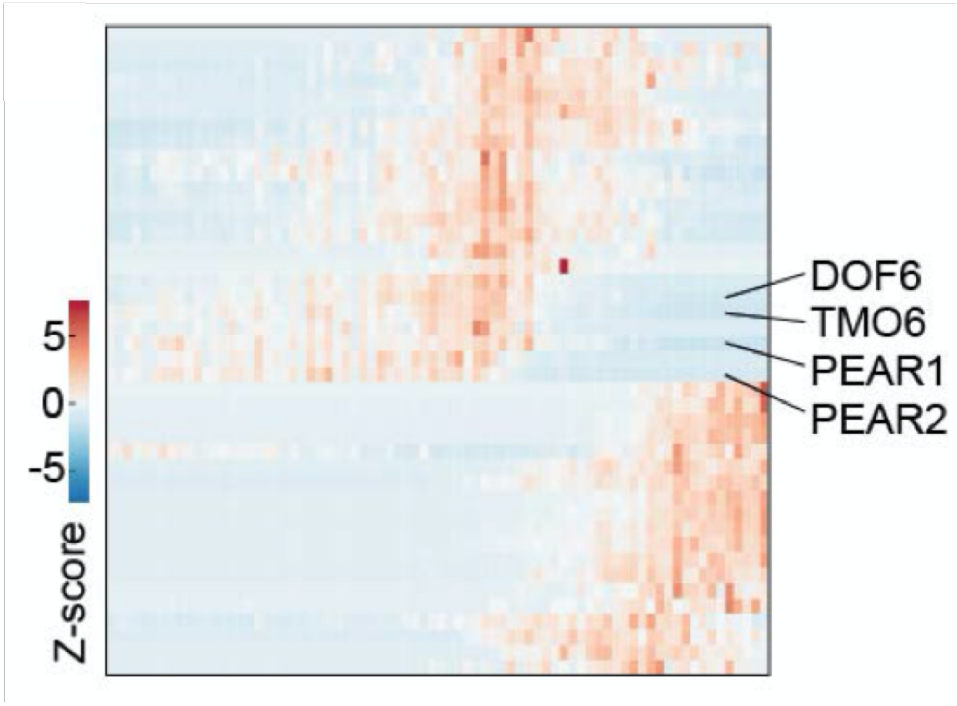


Figure 4.1: Expression heatmap reveals four PEAR genes among the earliest phloem specific transcription factors.

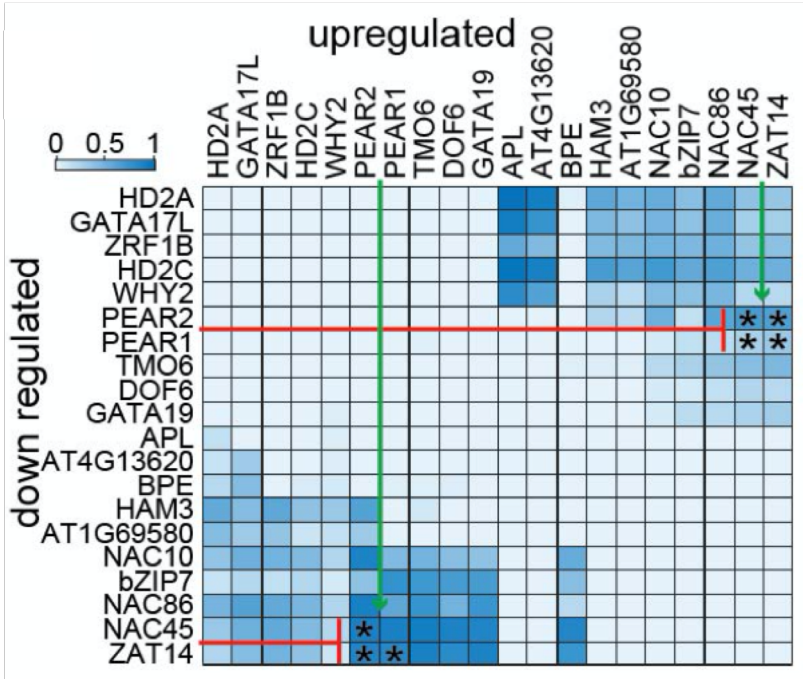


Figure 4.2: This heatmap shows significantly overlapping and oppositely regulated target sets of the 20 most important TFs from the GRN model. Colour intensity shows a fraction of overlapping target sets. The colormap represents significantly overlapping sets (Fisher Exact Test, if $p<0.05$, val=1) multiplied by the fraction of overlap. Asterisk indicates experimental validation of up and downregulated sets from TF OE in vivo.

The Gene Regulatory Network analysis shows that TFs along the trajectory appeared to repress each other's targets. i.e., the seesaw model and we used over-expression data in the plant to test validate it.

Chapter 5

PhenoPredict

This chapter describes another important application that uses data from gene expression to construct causality models. The aim of the machine learning strategy in this scenario is to create causal models for the phenotypes. In other words, the causality models are constructed to predict phenotypes from gene expression data. This is valuable because if, say, some gene g is over-expressed when plants have a positive phenotypic trait such as high yield, we want to infer what causes g to be over-expressed so that plants can be transformed to reach this desired trait. We address a case study in rice in which we were able to estimate both biomass and yield in two-month old plants on the basis of a model based on plants that were just a few weeks old.

5.1 PhenoPredict, a tree ensemble algorithm that predicts phenotype from gene expression data

Nitrogen (N) and Water (W) – two essential resources for plant growth – are increasingly limited inputs to modern agriculture. While agronomists have long-known that N-by-W interactions have a synergistic effect of on crop outcomes, the molecular basis for this remains unknown. The reason is that traditional experimental designs can determine either W- or N-response genes, not their interactions. To fill this knowledge-gap, our experimental colleagues use gene expression data from a novel N-by-W matrix expression design in order to uncover field outcomes.

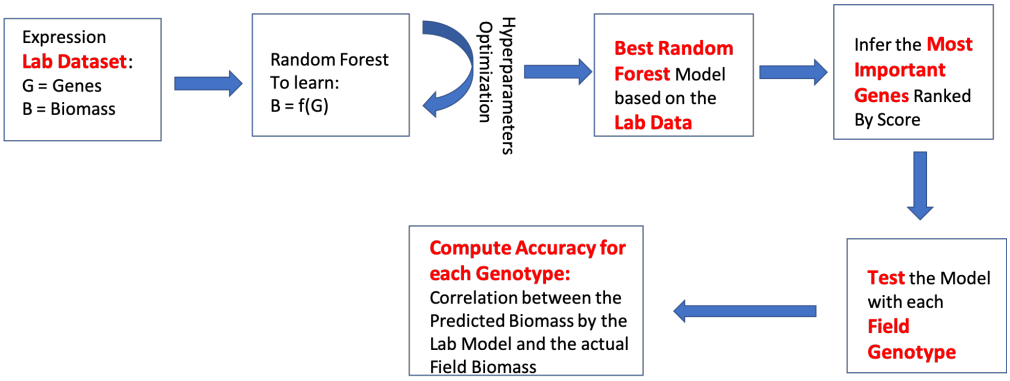


Figure 5.1: Pipeline for Modeling Biomass/Yield

PhenoPredict builds Random Forest-based models[15] using the N-by-W gene expression dataset. In this case study, we used PhenoPredict on the rice data from Nipponbare seedlings exposed to the 4x4 N-by-W matrix (16 conditions x 3 replicates) [83] to learn the model on rice seedlings, and then applied the learned model to predict both the biomass and yield of 19 rice genotypes in the field, at two months (mature plants) (Fig. 5.2, Fig. 5.3).

Encouragingly, the model learned on N-by-W gene expression data from Nipponbare rice seedlings, can predict biomass (Fig. 5.2) and yield (Fig. 5.3) in multiple rice genotypes of mature field grown plants. In addition, the list of TFs that are important to predict biomass in Nipponbare include genes that are present in most of the other genotypes. This indicates that the finding is applicable across different genotypes. Furthermore, included in the gene list are TFs whose gene expression levels can predict phenotypes. Among them are TFs validated by their reported function in heat stress[92, 51] as well as salt and dehydration stress[51] responses in rice.

The PhenoPredict pipeline was able to identify genes predictive of biomass/yield in mature plants from much younger plants. Specifically, the biomass model for Nipponbare at three weeks can be used on two-month old IR83383, PSBRC, IR74371_54, IR87707, PR106, Nipponbare, IR74371_70, IR64 (genotypes) to predict their biomasses (Figure 5.2) and on two-month old Palawan, IR20, IR83380, IR74371_70, PSBRC, IR83383, IR74371_54 (genotypes) to predict their yields (Fig. 5.3).

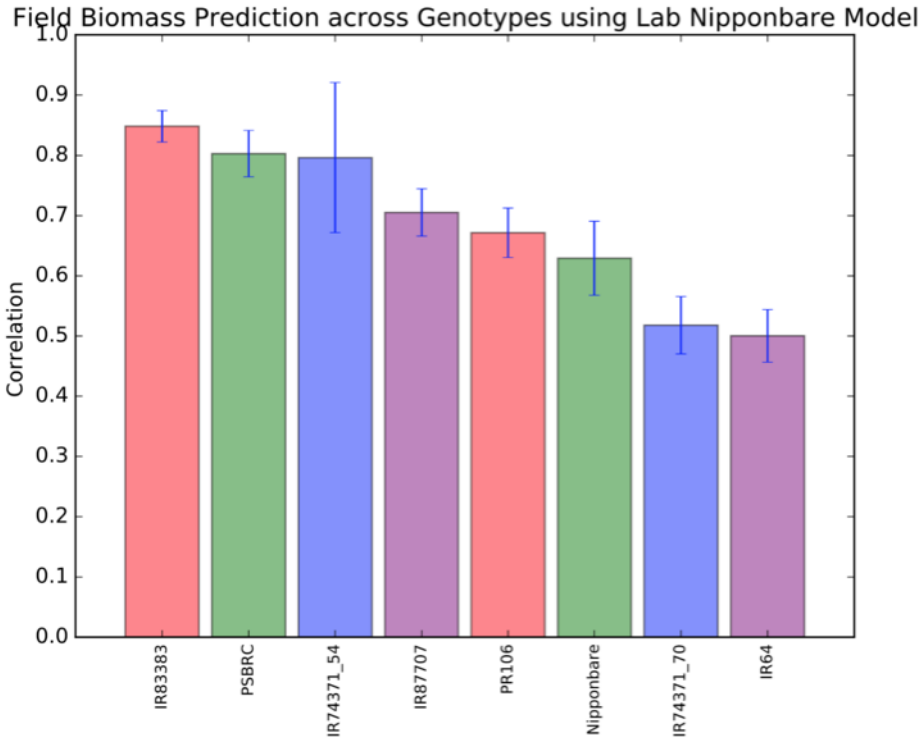


Figure 5.2: PhenoPredict models of gene expression -> phenotype learned using N-by-W response data in Nipponbare rice seedlings correlate with actual values of biomass and yield across rice varieties in the field. Top rice genotypes with predicted biomass using N-by-W data from Niponbarre seedlings to predict outcomes in 19 rice varieties in the field using data from [83]. The correlation is above 0.5 and standard deviation below 0.3. The y-axis is the correlation between the actual biomass values (of a given genotype) and the predicted values.

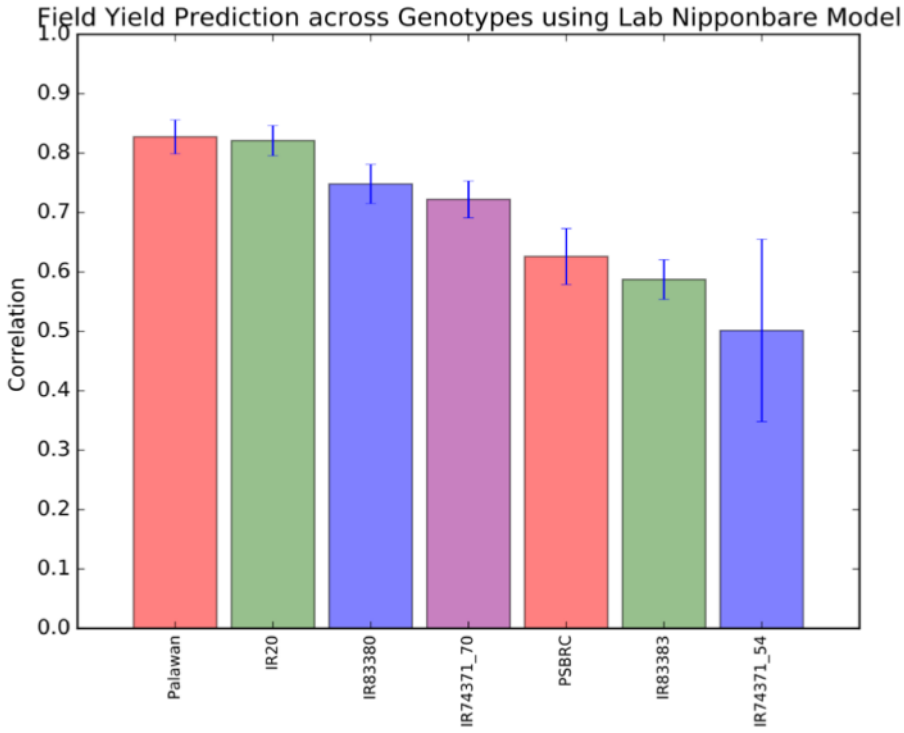


Figure 5.3: PhenoPredict models of gene expression -> phenotype learned using N-by-W response data in Nipponbare rice seedlings correlate with actual values of biomass and yield across rice varieties in the field. Top rice genotypes with predicted yield using N-by-W data from Niponbarre seedlings to predict yield outcomes in 19 rice varieties in the field using data from [83]. The correlation is above 0.5 and standard deviation below 0.3. The y-axis is the correlation between the actual yield values (of a given genotype) and the predicted values.

PhenoPredict with XGBoost

The Random Forest algorithm is based on fully grown decision trees built in parallel. By contrast, a decision-tree-based Boosting algorithm consists of weak learners, which are shallow trees built sequentially.

XGBoosting claims to provide a scalable, portable, and distributed Gradient Boosting implementation.

XGBoost fits a Regression Tree to the residuals, however, it does not use regular off-the-shelf regression trees, but a unique regression tree that we will call an XGBoost tree.

PhenoPredict uses the default method to build an XGBoost tree: each tree starts out as a single leaf or root node and all of the residuals go to the root, which is first node of the tree. Then, a quality score or similarity score for the residuals is calculated as the sum of the residuals squared, divided by the number of residuals plus λ:

$$Similarity\_Score = \frac{\sum_i^N {R_i}^2}{(N + \lambda)}$$

where R_i represents the residual of a given data point i; N is the total number of residuals or data points; λ is a regularization parameter that is intended to prevent overfitting the training data and reduce the prediction's sensitivity to individual observations.

After that, when a node is split, the question is whether or not the XGBoost algorithm is able to do a better job at clustering similar residuals if they are split into two groups.

The following formula for the variable *Gain* quantifies how much better the leaves cluster similar residuals than the root:

$$Gain = Left\_Similarity + Right\_Similarity - Root\_Similarity$$

At node splitting time, in a Random Forest the total reduction of variance is maximized, XGBoost, on the other hand, maximizes the *Gain*, so that for a given branch, the threshold that gives the largest *Gain* is used.

In order to prevent overfitting, an XGBoost tree is pruned according to its *Gain*

values and the γ regularization parameter, such that for a given node, if $(Gain - \gamma)$ is negative the branch is removed.

Hence, when λ is greater than zero the similarity scores are smaller and pruning becomes more likely, because the *Gain* values are smaller as well.

While setting γ equal to zero does not turn off pruning, setting Lambda equal to 0 turns off regularization.

XGBoost makes new predictions by starting with the initial prediction and adding the output of each tree, which is scaled by a learning rate η (another hyper-parameter):

$$Final\_Prediction = Initial\_Pred + \eta * Tree_1\_Pred + ... + \eta * Tree_N\_Pred$$

In summary, when building XGBoost trees, *Similarity Scores* and *Gain* are calculated to determine how to split the data. Then the tree is pruned by calculating the differences between *Gain* values and a tree complexity hyper-parameter γ: if the difference is positive then the tree is not pruned, if it's negative then pruning is done (i.e., the branch is removed). Afterwards, the output values for the remaining leaves are calculated as follows:

$$Output\_Value\_XGBoost = \frac{\sum_i^N R_i}{(N + \lambda)}$$

Lastly, λ is a regularization parameter, which affects pruning and output values when it is greater than zero, therefore the algorithm results in more pruning by shrinking the *Similarity Scores* and in smaller output values for the leaves.

We built an XGBoost model by performing an exhaustive hyper-parameters tuning search for λ, γ, η and other minor hyper-parameters.

As shown in Figure 5.4, we obtained comparable results with the ones presented in Random Forest (Figure 5.2). The overall accuracy and the number of top predicted genotypes are equivalent to the Random Forest version of PhenoPredict. The Random Forest and XBoost models both predict eight field genotypes well. The biomass Random Forest model for Nipponbare at three weeks can be used on the following eight two month-old genotypes, IR83383, PSBRC, IR74371_54, IR87707, PR106, Nipponbare, IR74371_70, IR64 (Figure 5.2), to predict their biomasses with significantly high accuracy. Further, six out of eight Random Forest high-accuracy predicted genotypes, i.e. IR87707, IR74371_54, PR106, PSBRC, Nipponbare, IR83383, are in common with the XGBoost well-predicted genotypes set (Figure 5.4).

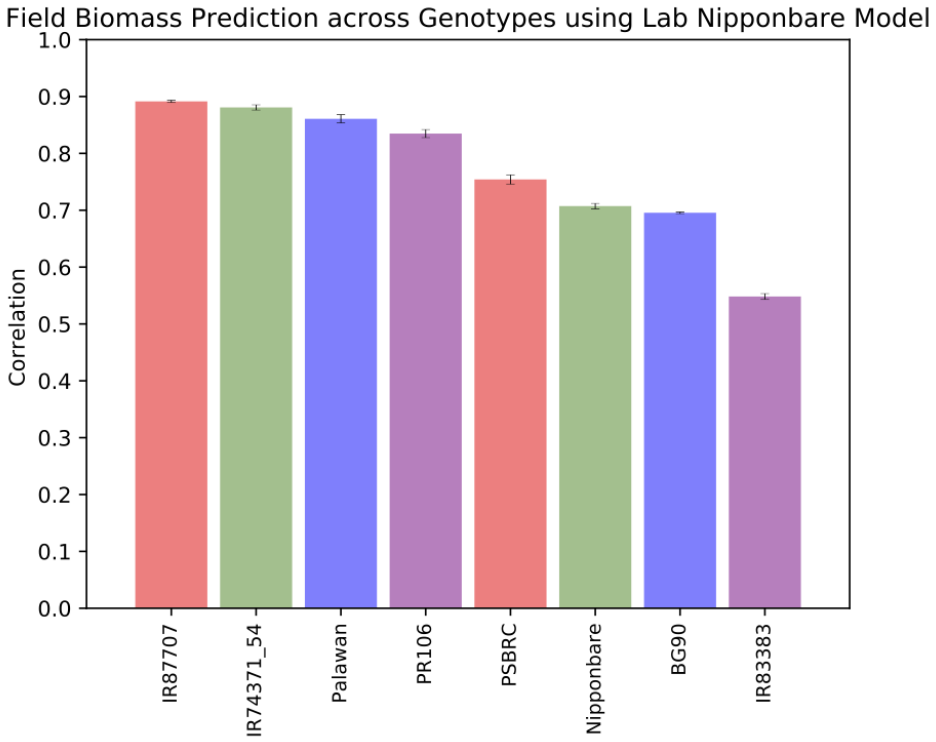


Figure 5.4: PhenoPredict with XGboost with the rice data: Top rice genotypes with predicted biomass using N-by-W data from Niponbarre seedlings to predict outcomes in 19 rice varieties in the field using data from [83]. The correlation is above 0.5 and standard deviation below 0.3. The y-axis is the correlation between the actual biomass values (of a given genotype) and the predicted values

Conclusion

As mentioned, the list of transcription factors that are shown to be important to predict biomass in Nipponbare include genes that are present in most of these other genotypes (Figure 5.5). Some of these already have a known role in positive phenotypes.

Our model both (i) suggests which transcription factors to test for at three weeks to predict high biomass and high yield at two months and (ii) which transcription

factors to over-express and which ones to repress in transformed plants to achieve higher biomass and yield.

number	msu7	rapdb	ranking	sign	gene	Validation_Score
6	LOC_Os03g50310	Os03g0711100	109	1	OsCOL10	3
12	LOC_Os01g07120	Os01g0165000	212	-1	OsDREB2A	3
16	LOC_Os05g45410	Os05g0530400	291	-1	SPL7	3
18	LOC_Os09g28354	Os09g0456800	324	-1	HSFB1	3
27	LOC_Os01g66030	Os01g0883100	435	1	OsMADS2	3

Figure 5.5: 30 TFs are present in the top 500 most important genes of the Lab model. Validation score column ranges from 0 to 3: 3 implies that a gene has solid experimental evidence from the literature on plant development and or stress tolerance

Chapter 6

Conclusions

The goal of this thesis is to provide a systematic framework for learning causality that is easy to use for biologists.

Towards achieving that goal, we focused on developing methods and tools for learning causality and predicting out-of-sample data using gene expression datasets.

This thesis tackles the problem of learning gene causality by constructing robust predictive models of out-of-sample data which embody accurate causal relationships.

In particular, we focused on our novel easy-to-use approach called OutPredict, which is a non-linear machine learning method based on an ensemble of regression trees for time series forecasting. It can incorporate steady-state data, temporal data and prior knowledge, as well as a variety of differential equation models for this purpose. OutPredict both predicts the future states of a given organism and gives a quantitative measure of the importance of a given transcription factor on a target gene.

OutPredict achieved high accuracy and significantly outperformed baseline and state-of-the-art methods.

In Chapter 4 I showed our application of OutPredict to single cell RNA-seq, so that we moved from inferring causality networks of Chapter 3 that describe the regulation of one type of cell to the environment to networks of influence in Chapter 4 that describe the differentiation of cell types. An important outcome was the identification of cell-type specific transcriptional regulators involved in cellular differentiation.

Finally, in Chapter 5 I have illustrated a variant of OutPredict called PhenoPredict which builds causality models to accurately predict phenotypes and then learning the effect of genes and TFs on phenotypes. Our PhenoPredict yields a model built on plants data at three weeks which then predicted very well biomass and yield across different genotypes in the field at two months.

The results presented in this thesis naturally open several interesting research directions. An exciting future work project is the combination of approaches to model cooperative modes of action where two or more TFs interact to target a gene; this would assume that there is a way to explicitly encode and device such interactions which are considered hyper-edges. Another improvement could be to utilize the extent to which binding and open chromatin assays to influence the selection of TFs to be branch points in the individual decision trees, since OutPredict can incorporate a variety of data sources. Finally, another interesting research problem is to extend our framework to allow cross-species inference via gene orthology.

Appendix A

Dynamic Factor Graph with Plant Model Organism

A.1 Temporal transcriptional logic of dynamic regulatory networks underlying nitrogen signaling and use in plants

[1] This study exploits time, the relatively unexplored 4th dimension of gene regulatory networks (GRNs), to learn the temporal transcriptional logic underlying dynamic nitrogen (N) signaling in plants [86] and our just-in-time analysis of transcriptome data uncovered a temporal cascade of cis-elements underlying dynamic N-signaling.

Nitrogen (N) - a nutrient/signal - is a core component of fertilizer used in modern

[1] Reference number [86], PNAS publication

agriculture to alleviate world-wide hunger ([62]). However, this comes at environmental costs, through excess nitrogen run-off due to inefficient N-use efficiency by crops ([88]). Thus, improving plant N-uptake, assimilation and utilization is highly desirable. With this goal, studies have attempted to capture and model the N-regulatory networks controlling N-uptake/assimilation ([71, 72, 48, 70]). Validation studies have identified several Transcription Factors (TFs) ([53, 42, 29, 4, 74, 66]) etc. as key regulators of N-signaling. However, we lack knowledge of the dynamics and temporal hierarchy of these known - and as yet unknown - TFs in controlling N-signaling, N-uptake/assimilation. A meta-analysis placed some known regulators within network modules ([40]). However, such correlation-based networks are unable to predict causality. By contrast, time-based machine learning approaches can predict the regulatory influence of TFs on their targets in the dataset and in out-of-sample data, the ultimate goal of systems biology ([48], [47], [58]).

In this study, we derived the temporal dynamics of N-regulatory networks by devising and combining several time-based approaches. First, our just-in-time analysis uncovered a temporal cis-element cascade underlying dynamic N-signaling. Second, we used a validated time-driven machine-learning approach, Dynamic Factor Graph (DFG) ([48], [47], [58]), to infer TF-target interactions in 2,174 N-response genes in shoots. Third, we pruned the inferred TF-target edges in this Gene Regulatory Network (GRN) using a precision cut-off threshold derived from experimentally regulated genome-wide targets of six novel regulators of N-uptake/assimilation - CRF4, SNZ, CDF1, HHO5/6, PHL1- validated herein. This pruned GRN predicts the influence of 155 TFs on 608 N-responsive genes. Fourth, to provide further support for the edges in the GRN, we used available TF-target

binding data (DAP-Seq) ([73]), also used to calculate a TF N-specificity index. This time-based GRN now reveals the temporal relationships of TFs previously validated in the N-response (e.g. NLP7/8([53], [26]), TGA1/4([42]), NAC4([29]), HRS1([4]), LBD37,38,39([74])). It also connects these known TFs with potential new TFs in the N-response cascade, including novel ones we validated herein - CRF4, SNZ, CDF1, HHO5/6, PHL1- to regulate a significant number of genes in the dynamic N-response, including 54% of nitrate uptake/assimilation pathway genes. Finally, we show that perturbation of CRF4, the earliest N-responsive TF in this GRN, affects genes and processes that result in altered nitrate-uptake, root development, and plant biomass, under low-N input conditions. Beyond these proof-of-principle examples, the pruned GRN of dynamic N-signaling we derived now provides the temporal transcriptional logic for 155 candidate TFs for perturbations aimed at improving Nitrogen Use Efficiency (NUE) with potential applications in agriculture. More broadly, these time-based approaches can be applied to uncover the temporal transcriptional logic for any biological response system in biology, agriculture or medicine.

A.1.1 A fine-scale time-course transcriptome of dynamic nitrogen signaling

Nitrogen (N) nutrient signal elicits dynamic responses in plant metabolism and development ([48, 40, 5, 78, 91, 3, 45]). However, most prior transcriptome studies assayed only one or two time-points following N-treatment ([71], [70], [40]),

or widely spaced time-points, not amenable to learning GRN causality ([45]). A previous study uncovered the very early (3-20 min) transcriptional response to nitrate-treatment in Arabidopsis roots ([48]). Herein, we captured early-to-late transcriptome responses (5, 10, 15, 20, 30, 45, 60, 90 and 120 min) to a N-supply (NO3- and NH4+) shown to elicit inorganic- and organic-N responses ([72]). Genes responding to N as a function of time (NxTime genes), were identified using a Cubic-Spline Model (FDR p-val<0.01) ([55]) (Fig. A.1 A&B). This analysis identified NxTime response genes in shoots (2,174 genes) and in roots (2,681 genes) (Fig. A.2C (Shoots: green bars), Fig. A.5B (Roots: brown bars). These NxTime gene sets are largely organ-specific, but share 778 genes, including 54 TFs (Fig. A.1C). These include many known N-responsive genes ([71, 72, 48, 70, 40]), and also 2,737 novel N-responsive genes (Fig. A.3, ([25])), due to increased sensitivity from RNA-seq and 511 genes absent on microarrays ([40]). We also captured new transient responses to N-supply, including the well-known N-regulator TF, NLP7 ([53], [26]) (Fig. A.14). Our dataset captures dynamic effects of N-signaling in shoots, on metabolism, RNA processing, photosynthesis ([8]) and circadian rhythm ([72]) (Fig. A.2, A.6, A.7, A.8).

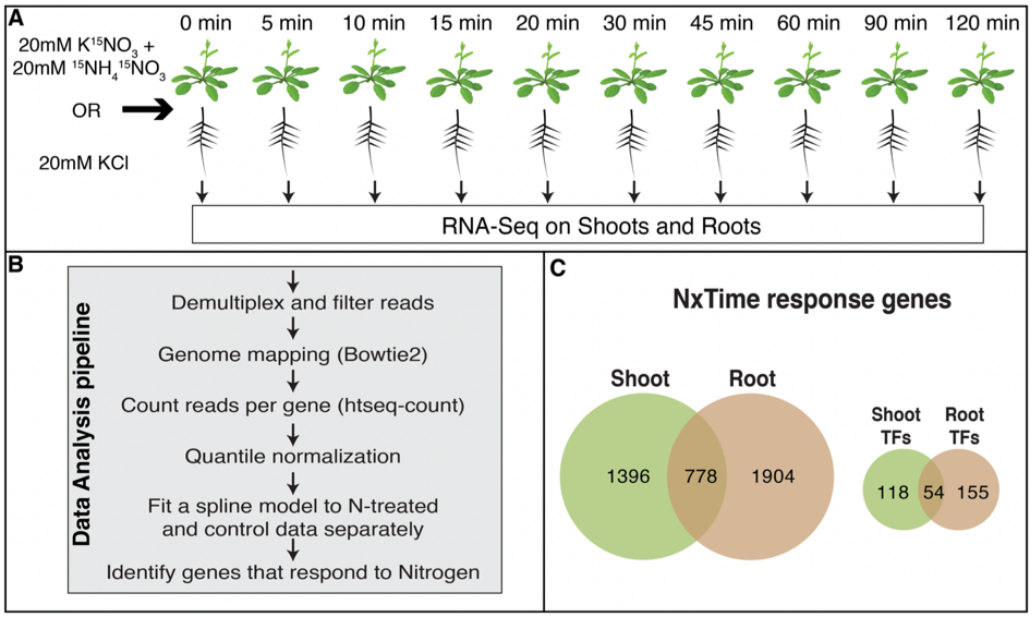


Figure A.1: **A fine-scale time-series profile of plant transcriptional changes in response to N- supply.** A. Three replicates of plants grown in a hydroponic system under low, but sufficient N conditions (1mM KNO3), were treated with either the N-supply in MS media (20 mM KNO3 + 20mM NH4NO3) or 20 mM KCl and harvested at time intervals 0, 5, 10, 15, 20, 30, 45, 60, 90, and 120 min after treatment. Shoots and roots from three independent Phytatray experiments were harvested separately at each time-point, and their transcriptome assayed using the RNA-Seq protocol on the Illumina sequencing platform. B. The resultant RNA-Seq data was filtered for quality and redundancy and converted into gene expression measures using the informatics pipeline shown. Genes responsive to the N-signal were identified by fitting the gene expression measures to a cubic spline model and testing for significant difference (FDR$<$0.01) between the N-treated and control fits (refer to Method section that describes Spline Analysis). C. The NxTime response gene sets have 778 genes in common, but also include genes that respond only in the shoot (1,396 genes) or root (1,904 genes). Similarly, the shoot and root N- responses also have shared and unique sets of Transcription Factors (TFs).

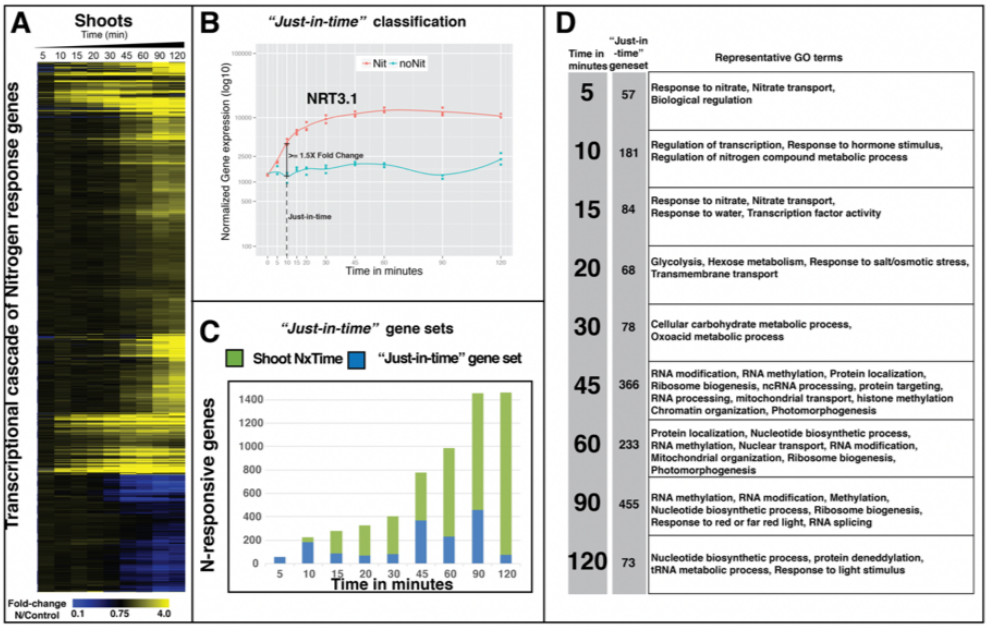


Figure A.2: **A fine-scale time-series profile of shoot transcriptional changes captures just-in-time responses to N-supply.** A. The transcriptional cascade triggered by N-signal perception shows a sequential activation and repression of 2,174 genes in shoots (NxTime genes), as identified by a Spline analysis [55]. B. Next, each NxTime gene is assigned to the first just- in-time point at which mean gene expression between +Nitrogen vs. No Nitrogen, changes by >=1.5 fold. C. The transcriptional response to Nitrogen in the shoots (i.e., size of NxTime genes) increases over time (Green bars). Blue bars = just-in-time gene sets identified using a classification algorithm to capture cohorts of genes whose expression is altered by the N-signal for the first time at that specific time-point. D. Next, each just-in-time gene set was analyzed by the BioMaps function in VirtualPlant [60] to identify overrepresented GO terms in each bin.

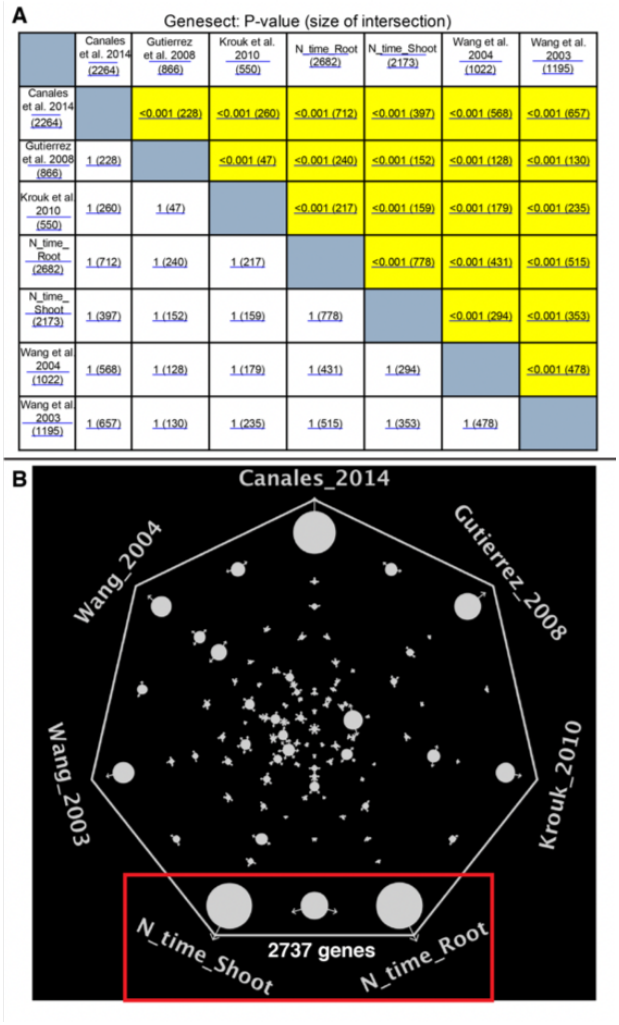


Figure A.3: **Fine-scale time-series captures known and novel genes in N-signal response.** A. Fine-scale N-response time-course in this study (N_time_Shoot and N_time_root) captures the known N-response genes from previous studies ([71], [70], [40], [72], [48]). The display uses the GeneSect function in VirtualPlant [60] to calculate the significance of the gene intersect. B. Finds a novel set of 2,737 N-response genes unique to our new N_time_Shoot and N_time_root, as visualized using the SunGear function [25] in VirtualPlant [60].

A.1.2 Just-in-time analysis uncovers a temporal cascade of cis-regulatory elements and biological processes in response to N-supply

To uncover the regulatory cascade underlying dynamic N-signaling, we implemented a just-in-time (JIT) analysis. This JIT analysis bins NxTime genes, based on the first time-point at which its mRNA levels are affected by N-signaling (Fold-Change (FC) >= 1.5) (Fig. A.2B, Fig. A.2C, blue bars). We then identified overrepresented known cis-motifs ([73], [7], [9]) in each JIT bin, using a hypergeometric distribution on a genome-wide promoter background ([89]). This analysis uncovered a temporal cascade of over-represented cis-regulatory motifs (e-val <0.05) in the promoters of genes first responding to N-signaling at each JIT point (Fig. A.4A). The set of enriched cis-elements are different between the JIT sets of shoots (Fig. A.4A) vs. roots (Fig. A.5C). The temporal enrichment of unique cis-element motifs in shoots is particularly noticeable at the 10, 15 and 20 min JIT points (Fig. A.4A). Conversely, certain cis-element motifs - such as SORLIP2 and TELO-box - are over-represented at consecutive JIT sets (Fig. A.4A). This JIT analysis also uncovered a temporal cascade of enriched Gene Ontology (GO) terms enriched in each JIT gene set in shoots (FDR adjusted p <0.01) (Fig. A.4B, Fig. A.2D). The early JIT gene sets (5-15 min) are significantly enriched in genes related to N-uptake/assimilation. Intermediate JIT gene sets (20-30 min) are enriched in energy generation. The later JIT gene sets (>=45 min), are enriched in genes for metabolic and developmental processes (Fig. A.2D, Figs. A.6, A.7, A.8, A.13). Overall, JIT cis-element and GO analysis, implicates a cascade of associated TFs regulating largely non-overlapping sets of genes at consecutive JIT time-points in

N-signaling (Fig. A.4). However, the current cis-motif datasets ([73], [7], [9]), are generalized for TF families, and cannot associate individual TFs with specific target genes. We thus associated specific TFs with targets in the NxTime cascade by using a time-based network inference method described below.

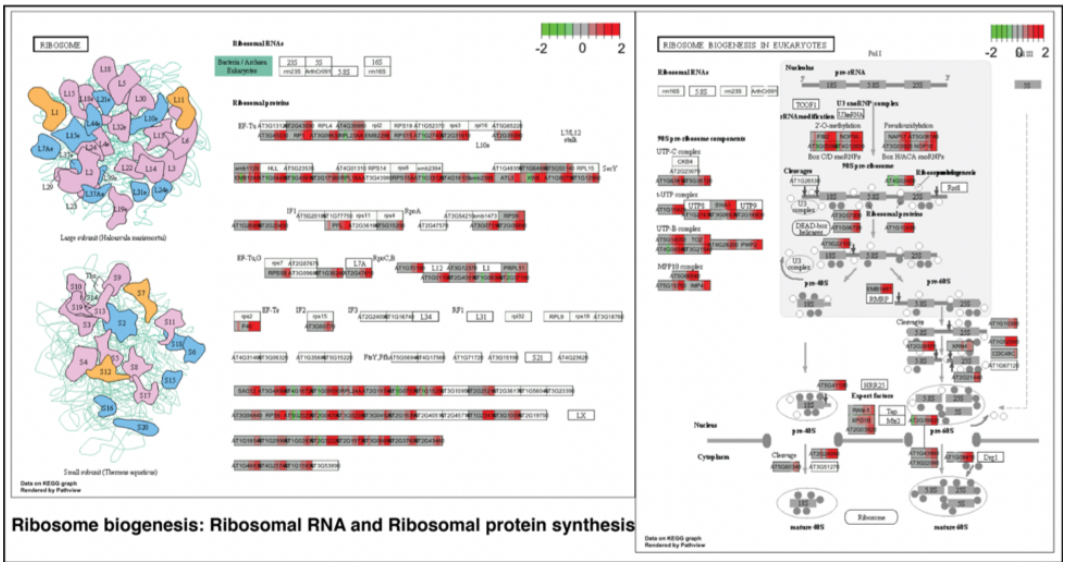


Figure A.6: N-signal in shoots stimulates multiple components of the ribosome biogenesis pathway [87]. The N-signal induces a coordinated up-regulation of mRNA for ribosomal RNA subunits and ribosomal proteins, starting at 30 minutes after the initial N-signal.

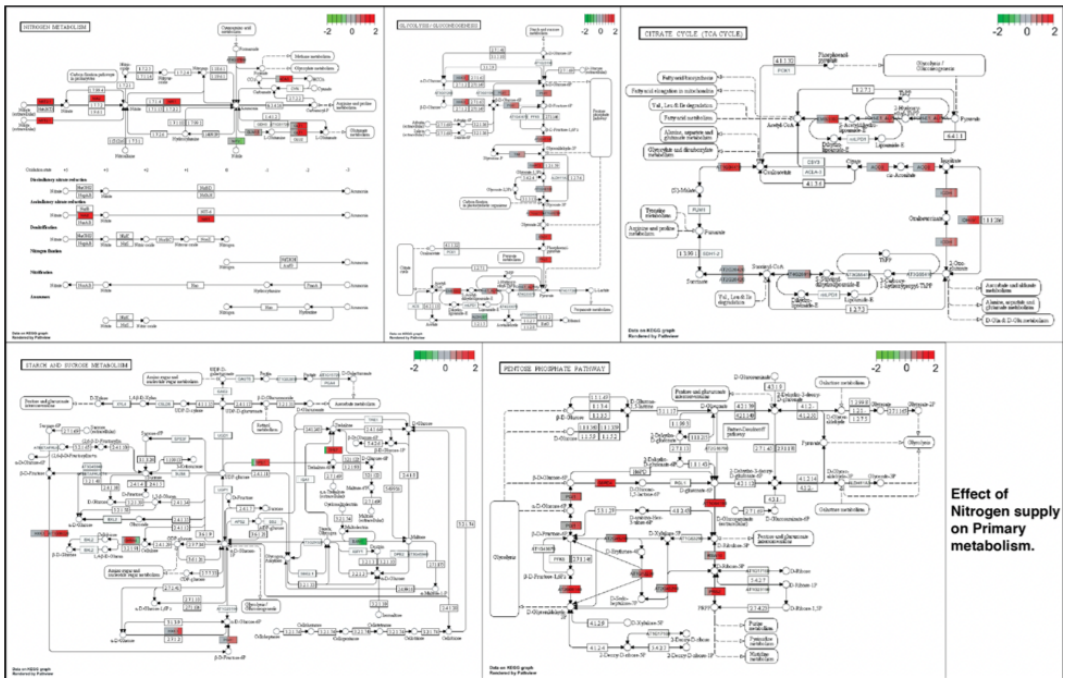


Figure A.7: N-signal induces multiple pathways [87] in plant primary metabolism within 5 minutes of N-supply in shoots. These pathways are either directly involved in nitrate uptake or in providing Carbon skeletons and/or energy for nitrate assimilation.

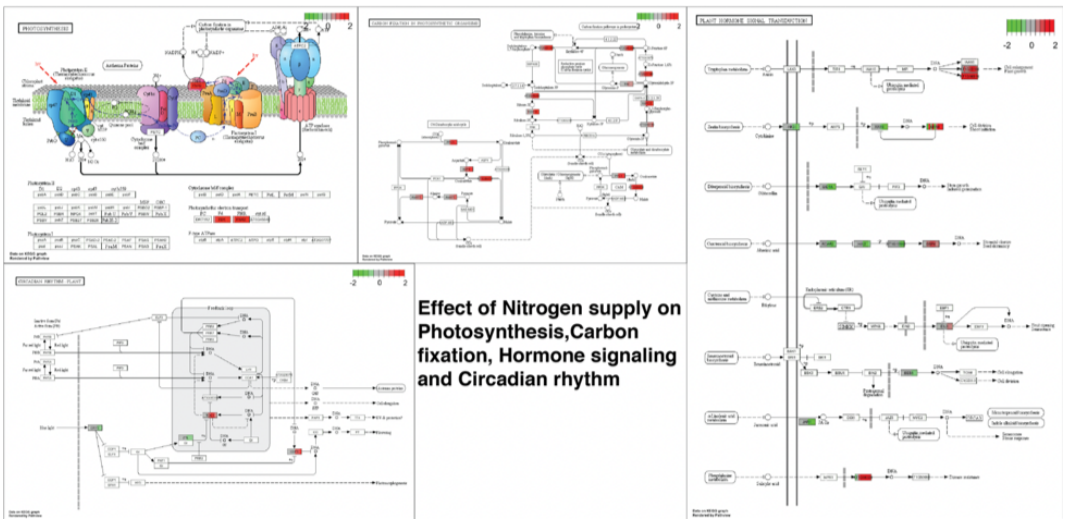
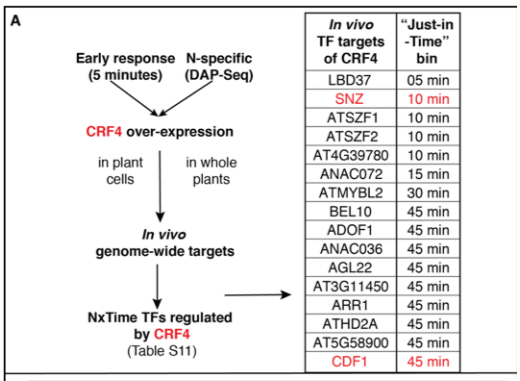


Figure A.8: N-signal response affects processes beyond cellular metabolism in shoots. Multiple steps in the carbon fixation, photosynthesis, hormone signaling and the circadian rhythm pathways [87] are altered in response to the N-supply in shoots. These changes happen later in the N-signal response (i.e., >30 minutes), compared to the changes induced by the N-signal in primary metabolism (5-20 minutes) (Fig. A.7).

A.1.3 Assigning a N-specificity index to TFs in the dynamic N-response cascade

Our time-course captures 172 TFs responding to N-supply within two hours. To identify TFs that play a specific role in N-signaling, we computed a N-specificity index, based on available TF-target binding data ([73]). For each NxTime regulated TF with genome-wide binding data (40 TFs), we tested if the proportion of its genome-wide targets ([73]) in the NxTime shoot genes are significantly over-represented, relative to the proportion of all the TF-bound targets in the genome. This identified 19 TFs with a highly significant N-specificity score (p-val$<$0.05) in shoots. These N-specific TFs include four validated regulators of the N-response

(NLP7([53]), TGA1/4([42]) and NAC4(9)), and 15 novel TFs whose targets are enriched in N-signal responsive genes in shoots. We note that this N-specificity calculation is limited to TFs with TF-Target binding data for 529 TFs currently in the DAP-seq database ([73]). However, this N-specificity calculation may be applied to any TF, with known genome-wide targets as we show with SNZ and CDF1 (Fig. A.9B), as detailed below.



In vivo TF targets of CRF4	"Just-in-Time" bin
LBD37	05 min
SNZ	10 min
ATSZF1	10 min
ATSZF2	10 min
AT4G39780	10 min
ANAC072	15 min
ATMYBL2	30 min
BEL10	45 min
ADOF1	45 min
ANAC036	45 min
AGL22	45 min
AT3G11450	45 min
ARR1	45 min
ATHD2A	45 min
AT5G58900	45 min
CDF1	45 min

B

Genome-wide targets	Just-in-Time	Over-represented GO terms (Full lists in Tables S8-10,S27-30)	NxTime genes (2174)	Shoot N-specificity*
CRF4 (2496)	5 min	Nitrate Assimilation Ribosome Biogenesis	p<7e-37 (371)	p<2.9e-60
SNZ (1753)	10 min	Nitrate Assimilation Signal Transduction	p<3e-8 (199)	p<2.3e-14
CDF1 (3193)	45 min	Nitrogen cycle metabolic process Rhythmic process	p<2e-23 (397)	p<1.9e-39

C

Genome-wide targets	Just-in-Time	Over-represented GO terms (Full lists in Tables S8-10,S27-30)	NxTime genes (2174)	Shoot N-specificity*
HHO5 (1540)	10 min	Phosphate metabolic process Nitrate transport	p<2e-11 (190)	p<8.8E-10
HHO6 (2298)	45 min	Phosphate metabolic process	p<1e-7 (111)	N/S
PHL1 (3247)	60 min	Regulation of nitrogen compound metabolic process	p<4e-17 (137)	N/S
TGA1 (1403)	15 min	Defense response Response to nitrogen compound	p<1e-3 (79)	N/S

Figure A.9: **Novel TF regulators - CRF4, SNZ, CDF1, HHO5, HHO6 & PHL1- of NxTime genes in shoots.** A. CRF4 was chosen for initial functional validation in planta, as it responds early to N-signaling (JIT: 5min) and has a high N-specificity index. In planta ([88]) and shoot cell-based transient TF perturbation assays [13] identified 16 TFs that are regulated by NxTime and by CRF4. From this set of CRF4 targets, SNZ (JIT:10min) and CDF1 (JIT:45min) were chosen for further validation by TF perturbation in shoot cells using the TARGET system [13]. B. Genome-wide regulated targets of CRF4, SNZ and CDF1 were validated by TF over-expression in plants [1] andor shoot cells [13]. C. Independently, genome-wide regulated targets of four additional TFs in the pruned GRN - HHO56, PHL1 and TGA1- were identified in shoot cells using the TARGET assay [13]. Genome-wide regulated targets of all of these seven TFs (Panels B and C), show a significant overlap with the NxTime gene set in shoots. Six of these seven TFs show over-representation of GO terms related to the Nitrogen assimilation process. Further, four novel TFs validated here (CRF4, SNZ, CDF1 and HHO5) also show a high N-specificity of regulated genes in shoots. Note: The N-specificity listed in Panel C is estimated from the regulated genome-wide targets of these four TFs in the shoot cell TARGET assays. By contrast, the N-specificity index shown in Fig 3 was estimated from genome-wide TF-target binding (DAP-Seq) in vitro [73].

A.1.4 CRF4 - the earliest TF in the N-signaling GRN - regulates N-uptake and N-use in planta

The pruned DFG network - refined by TF-target binding data - places CRF4 at the top of the N-signaling cascade (Fig. 2.6), based on its early response (5 min JIT) and its GRN connections. Indeed, our validation studies support the early and specific role of CRF4 in mediating the dynamic N-response GRN in planta. Inducible expression using a CRF4-OX transplanta line ([1]), reveals that CRF4 controls a highly significant number of NxTime genes, spanning early and later just-in-time gene sets (Fig. 2.5B&D). Impressively, CRF4 directly or indirectly regulates 1/3 of the genes in the N-uptake/assimilation pathway (21/65), including seven N-uptake genes (Fig. 2.4). In planta CRF4 targets are also enriched in N-metabolic processes and translation (in shoots), and response to nitrate and root development (in roots) (Fig. 2.5C). Moreover, these CRF4 mediated changes in gene regulation affect N-uptake and use in planta (Fig. A.10). CRF4-OX over-expression results in significantly lowered shoot biomass ($p<1e\text{-}5$) (Fig. A.10C), primary root length, and number of lateral roots, under low-N conditions (Fig. A.11A &B), where the high-affinity N-transporter, NRT2.1, is the major functional nitrate-uptake system [93]. Further, repression of NRT2.1 in shoots and roots of CRF4-OX plants (Fig. A.10A, A.11C), leads to lower rates of nitrate-uptake under low-N-conditions in the CRF4-OX line ([1]). Using 15NO3 tracer [76], 15NO3-uptake was significantly reduced in the induced CRF4-OX over-expression line, at levels comparable to the nrt2.1 mutant impaired in high-affinity nitrate uptake [75], when compared to un-induced CRF4-line and wild-type controls, under low-N conditions (2-way ANOVA, with Tukey HSD analysis) (Fig. A.10B). These results validate the im-

portant role CRF4 plays in regulating N-uptake/use - acting either directly, or indirectly through its downstream TFs, such as SNZ and CDF1 (Fig. A.10A).

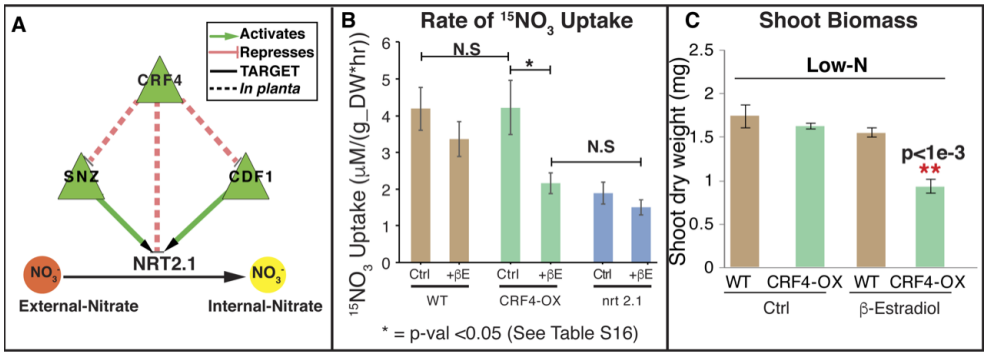


Figure A.10: **CRF4 overexpression represses high-affinity nitrate uptake and biomass in planta.** A. CRF4 overexpression via β-estradiol ($+\beta$ E) induction ([1]) represses SNZ, CDF1 and NRT2.1 (Fig. A.11C). SNZ and CDF1 over-expression in shoot cells ([13]) induces NRT2.1 expression. CRF4 over-expression in low-N (1 mM NO3) conditions significantly reduces; B. the rate of nitrate 15NO - uptake, and C. Shoot biomass in planta (Fig. A.11A).

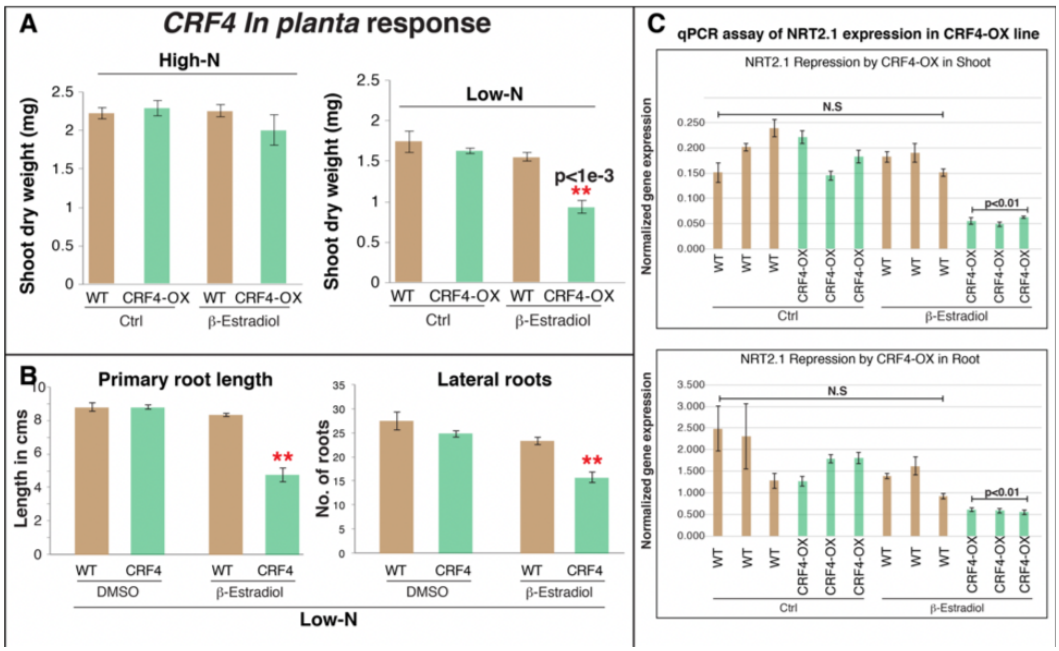


Figure A.11: Conditional CRF4 over-expression in planta leads to changes in shoot biomass, root growth and NRT2.1 expression. A. Conditional and sustained induction of CRF4-OX in plants grown for 7 days under low-N conditions (1mM N), results in significantly lower shoot biomass specifically under low-N (Tukey test). This growth retardation in CRF4-OX is specific to low-N, is and is not observed in high-N conditions (30mM N). B. The induction of CRF4 over-expression by -Estradiol [1] also resulted in a reduction of primary root length and the number of lateral roots, under low-N conditions (1mM N). C. Q-PCR assays were used to measure the expression levels of NRT2.1 in shoots and roots of the CRF4-OX line and wild-type plants, under low-N conditions, in the presence/absence of the b-Estradiol. CRF4 over-expression, induced by b-Estradiol [1] represses NRT2.1 expression in roots of whole plants, as determined by a 2-way ANOVA analysis followed by TukeyHSD.

A.1.5 A fine-scale time-course and GRN establishes the temporal hierarchy of known and novel N-signaling regulators

Next, we used the DFG network inference method ([48], [58]), to derive GRNs that reveal the transcriptional logic underlying dynamic nitrogen (N) signaling in shoots. The resulting N-response network pruned for precision now places 155 N-responsive TFs in shoots in a temporal hierarchy (Fig. A.12), and predicts their likely temporal interactions. For example, the 12 TFs that respond earliest to the N-signal in shoots (JIT=5 min) (Fig. A.12), include TFs previously validated in the N-response: LBD37/38/39([74]) and HRS1([4]), and a novel early TF validated herein; CRF4 (Fig. 2.6). We note that some of the earliest steps of N-signal transduction are also likely to occur via post-translational modifications ([5]) or changes in TF localization, as shown for NLP7 ([53]). This pruned network, is further supported by TF-target binding data, - and reveals a set of 15 novel TFs that are specific to the N-response (Fig. 2.6, TFs in red). This establishes the power of applying de novo GRN inference approaches to expression data sets, as shown for GRNs mediating environmental responses in rice ([65]), and drought responses in Arabidopsis [85].

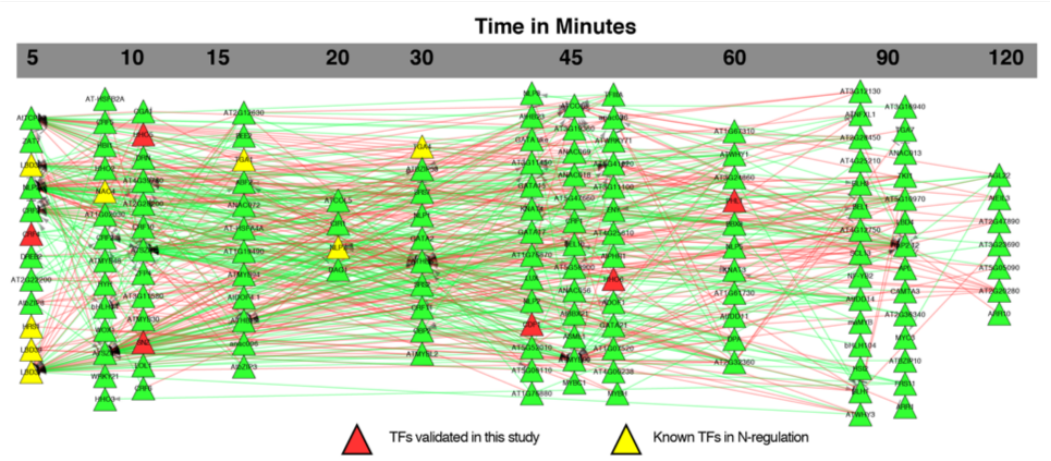


Figure A.12: Pruned DFG network predicts the temporal interactions of 155 Nx-Time responsive TFs in shoots. The pruned DFG N-regulatory network of shoots places TFs in a temporal hierarchy and predicts the regulatory interactions between them. The currently known TF regulators of N-signal response (e.g., NLP7, HRS1, TGA1/4, LBD 37/38/39, NAC4) are highlighted in yellow, while the six novel regulators (CRF4, SNZ, CDF1, HHO5, HHO6 and PHL1) of N-signal response functionally validated herein are shown in Red. The activation edges are shown in green while inhibitory edges are shown in red.

A.1.6 CRF4 - the earliest TF in the N-signaling GRN - regulates N-uptake and N-use in planta

Our time-based N-regulatory network revealed CRF4 as a novel early player in mediating the N-signaling response. Indeed, our genome-wide target studies and phenotypic analysis support the key role CRF4 plays in mediating nitrate uptake and use in planta (Fig. A.10, Fig. A.11). In response to N-supply, CRF4 represses genes in the N-assimilation pathway, including the high-affinity nitrate transporter NRT2.1 (Fig. A.11C), which is repressed under high-N conditions in wild-type [75]. Additionally, we validated two downstream TF targets of CRF4, and found that SNZ is largely an activator, while CDF1 activates or represses genes in the

N-assimilation pathway (Fig. 2.4). CRF4 targets in shoots include ribosomal proteins, induced within 30-45 min of N-supply (Fig. A.2D, Fig. 2.5C). In roots, CRF4 regulates nitrate uptake and root development processes, consistent with the in vivo phenotypes (Fig. A.10, Fig. A.11 & 2.5C). N-signaling is a novel role for CRF4, whose only previous description was the role it plays in the cold response [69]. In addition, we discovered that 11/12 members of the CRF family [69] are N-responsive, including eight in shoots (CRF1-6, CRF10 & CRF11), and three (CRF3-4, CRF11) in roots. This highlights a potential new role of the CRF family in linking the N-response and cytokinin signaling ([77]). Our study also identifies multiple novel TFs that link nitrogen and phosphate responses (HHO5/6, PHL1), as previously shown for HRS1([4]).

In addition to discovering novel TFs in the N-response network, the transcriptional logic of N-signaling uncovered herein can also suggest the temporal mode-of-action for TFs and combinatorial TF experiments which will be valuable for the global goal of enhancing NUE. More broadly, our time-centric approach that uses fine-scale time-course data to fuel causal network inference, can now be applied to understand any stimulus-driven gene regulatory network in any organism. Moreover, the analysis approaches we described - just-in-time and N-specificity index - can be used to uncover the regulatory structure and signal-specificity in any time-series transcriptome datasets. When coupled with genome-wide TF-target binding data (e.g. ChIP-Seq, DAP-Seq([73])) and other layers of genome-wide dynamic interaction data (e.g., chromatin accessibility maps([9])), the approach employed in our time-based study can identify key molecular players, their hierarchy and other emergent network properties in any complex transcriptional regulatory system in biology, agriculture, or medicine.

To summarize, Nitrogen (N) - a key nutrient/signal - regulates dynamic plant processes including circadian rhythm ([72]) and root-foraging ([48, 40, 5, 78, 91]). However, the underlying temporal mechanisms are unknown. Our just-in-time analysis uncovered discrete waves of transcriptional responses to N-signaling in shoots (Fig. A.4). For example, we confirm and extend the role of N-signaling as an input to the circadian clock in plants ([72]). N-signaling regulates TFs in the circadian clock, inducing TOC1 and CDF1, and repressing ZTL within 20-45 min after N-supply (Fig. A.8). Overall, the shoot NxTime gene set shows significant enrichment for genes with peak expression at pre-dawn ([84]).

A.1.7 Just-in-Time analysis of time-series transcriptome data

Each NxTime gene was assigned to the first time-bin at which gene expression in N-treated samples is >=1.5 fold of control (Fig. A.2B). Each just-in-time gene set (Fig. A.2C, blue bars), was analyzed to identify over-representation of cis-regulatory motifs (FDR E-val <0.05, Elefinder [7]) and such cis-elements were hierarchically clustered (Fig. A.4A). Just-in-time gene set were also analyzed to identify over-represented GO-terms [60] (Fig. A.4B, Fig. A.2D). The normalized expression level of the N-response genes in shoots (2,174 genes) and roots (2,681 genes) at each of the time-points assayed (0-120 min) was used to calculate the fold-change of expression between the N-treated samples and the controls (KCl). For just-in- time analysis, each gene was then assigned to the first time-bin at

which the fold-change of expression is >= 1.5 fold (See Fig. A.2B). The promoters of all genes in each just-in-time gene set (Fig. A.2C, blue bars), were then analyzed to identify over-representation of cis- regulatory element motif (FDR corrected E-val < 0.05) using an online search tool (Elefinder [89]). Cis-regulatory motifs that are rare in the genome were filtered out to remove spurious associations by requiring that for each just-in-time bin every over-represented cis-motif must be present in at least 5 or more promoters in that gene set. The resulting matrix of over-represented cis-elements in genes at each just-in-time points was hierarchically clustered and visualized using MeV [28] (Fig. A.2A). Separately, all genes in each just-in-time gene set were analyzed by the BioMaps function in VirtualPlant [60] to identify over-represented GO-terms in each bin (Fig. A.4B and Fig. A.2D).

A.1.8 Nitrogen-specificity index for TFs in the GRN

For each of the 40 NxTime TFs with in vitro TF-target genome-wide binding data [73, 44], we retrieved genome-wide targets in shoot NxTime set. The TFs with a significantly higher proportion of targets in the NxTime set relative to their genome-wide distribution (one-tail t-test, p-val <0.01), were accepted as being specific to the N-signal. Of the 172 TFs that respond to N-signal in the shoot, DAP-Seq in vitro TF-target genome-wide binding data [73, 44] is available for 40 of these TFs. For each of these TFs, their genome-wide targets were retrieved from the Plant Cistrome Database [73, 44]. We next obtained the subset of each TFs target in the N-signal response genes, by intersecting the genome-wide targets of each TF with the NxTime signal response genes in shoots (2,174 genes). For each TF, the proportion of its targets in the genome was calculated as $p_g = T_g/G_g$

where T_g is the total number of TF targets in the genome and G_g is the total number of genes in the genome. Again, for each TF, the proportion of its targets in the N-signal response gene set was calculated as $p_n = T_n/G_n$ where T_n is the total number of TF targets in the N-signal response gene set, and G_n is the total number of genes in the N-signal response gene set. The significance of each TF to the N-signal was then tested by a one-tailed t-test under the null hypothesis $p_n = p_g$. The TFs with a significantly higher p_n than p_g (p-val <0.01) were accepted as being specific to the N-signal.

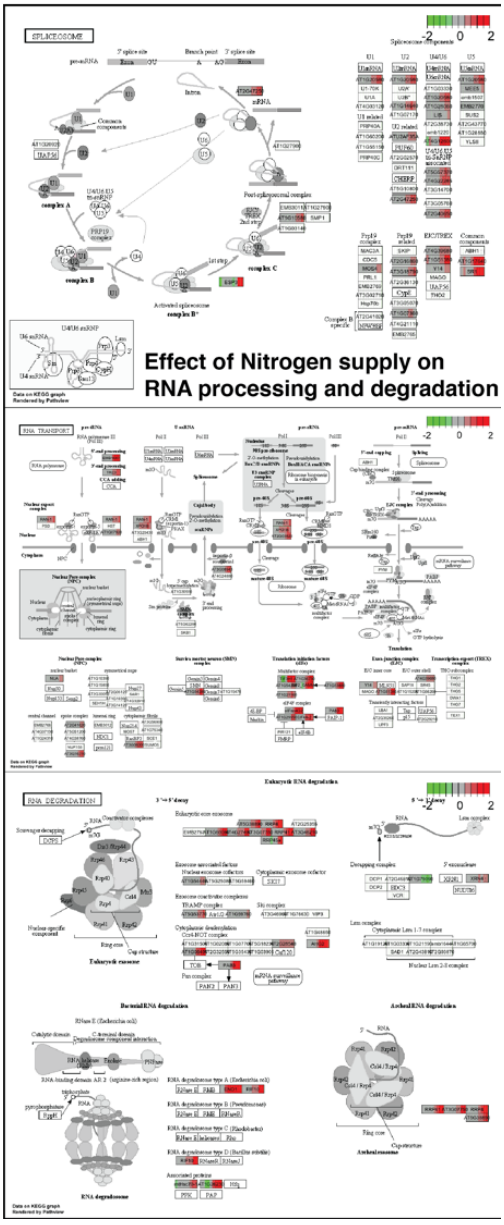


Figure A.13: N-signal alters the expression of various components in the RNA processing and degradation machinery in shoots. A novel observation of this study is the effect of N-supply on the molecular machinery required for proper processing of mRNAs and their degradation in shoots. Molecular components of both these pathways are upregulated, implying an increase in the mRNA turnover within the plant.

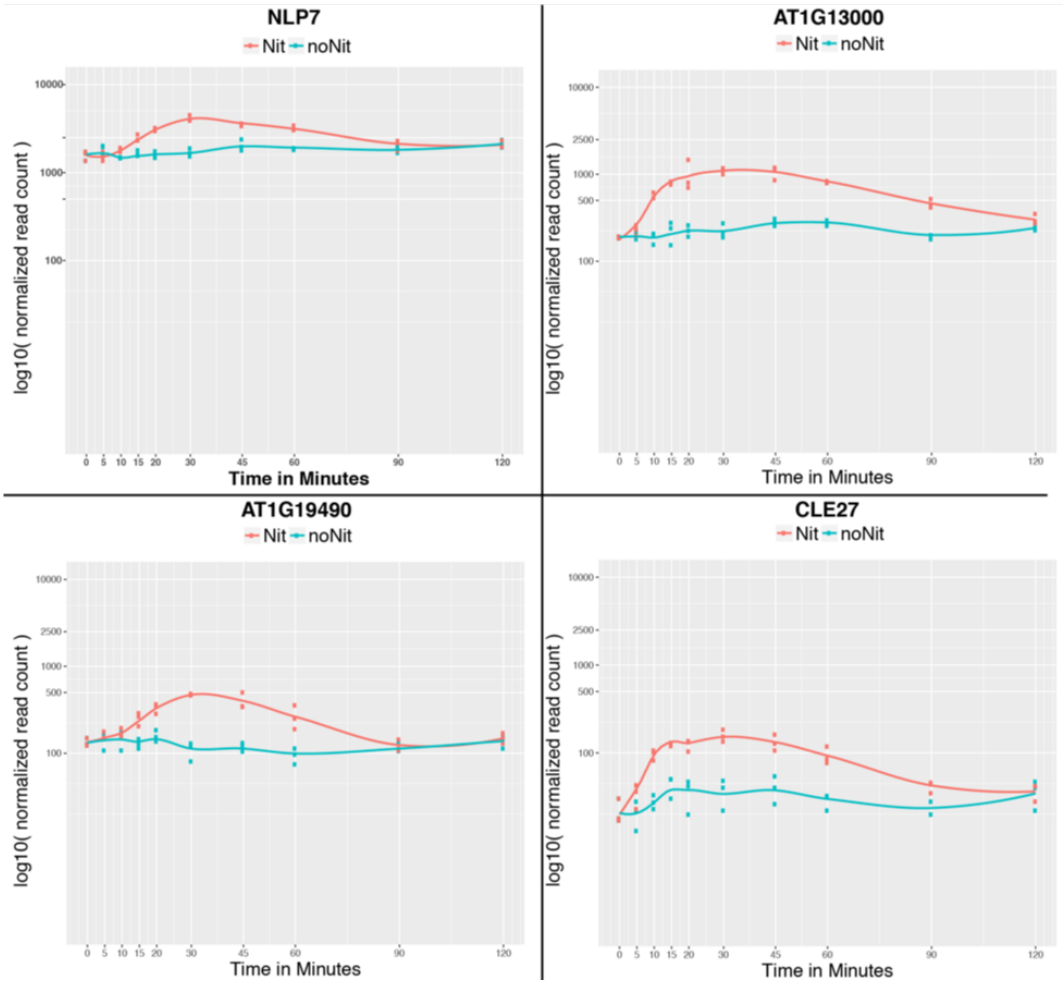


Figure A.14: Spline analysis of time-series transcriptome captures transient changes in N- regulated gene expression in shoots. Transient changes in N-regulated gene expression are generally missed in end-point measurements. Genes shown here to be N-regulated at earlier time- points would not be detected as N-responsive, if assayed only at 2 hours after N-signal. For example, NPL7 [19], a major player in the N-response was not previously known to be transcriptionally regulated by N-supply at these early time-points. Genes responding to nitrogen significantly (FDR adjusted p-val <0.05) over the time-series N-response data were identified by fitting a Cubic Spline Model (df=5) to the N-treatment and Control samples, using the lmFit function in the Limma R package [55] and visualized using ggplot2 [36]

A.2 Network Walking charts transcriptional dynamics of nitrogen signaling by integrating validated and predicted genome-wide interactions

[2]Across biology, a great deal of effort is being invested in generating gene regulatory networks that are able to accurately predict future states and identify regulatory hubs that can be manipulated to achieve desired phenotypes. Key to accomplishing this goal is linking the earliest responses to stimuli in the cells that perceive the signal to the downstream consequences of that signal at the tissue and organism level. However, the experimental tools most commonly used to validate the accuracy of these networks in vivo are prohibitively time-consuming and expensive to scale to the necessary level. A proliferation of TF binding data from in vitro methods has been helpful, however the results lack cellular context and cannot inform what effect the TF-target interaction has on gene regulation. In this chapter, we have used data which comes from a cell-based assay adapted to increase the throughput of identification of direct regulated TF targets [16]. This allows us to successfully chart and validate a temporal path in the nitrogen response gene regulatory network that links early TF-triggered events to downstream effects. This assay can be done on cells isolated from a specific tissue of interest, and can capture important transient events that are often missed in whole tissues. Herein, we use this assay to identify direct targets of 33 TFs that respond early to nitrogen treatment in Arabidopsis. Our results confirm the roles for 6 known regulators of nitrogen signaling and validate an additional 14 novel TFs in this important pathway, including information on their temporal course of action. Using a network we built

[2]Reference number [16], Nature Comm publication

from integrating direct targets identified for these 33 nitrogen-early response TFs with existing datasets, we demonstrate how the resulting TF-target connections can be used to generate new biological insights. Using all available cis-binding motifs for Arabidopsis and a recently published fine-scale nitrogen time-course, we identify families of TFs that work together, and extend our results to predict targets for an additional 145 nitrogen responsive TFs in Arabidopsis roots. Finally, we present our Network Walking approach using TGA1, a well-known TF in the nitrogen-response in which we chart a network path to connect the indirect targets which respond to a TF only in planta back to that TF via intermediate TF_2s. Our approaches for identifying direct regulated targets of a TF in cells isolated from a specific tissue type and integrating datatypes via Network Walking are easily adapted to other eukaryotic systems. This enables researchers to study any signaling network of interest and identify the temporal hierarchy of TF-target interactions. Our Network Walking is a combination of novel and scalable experimental/computational approaches to infer Temporal path in gene networks. In other words our approach is used to connect direct TF targets identified in cells (this is done using the wet lab protocol) to the indirect and downstream targets identified only in planta (this is done used our computational approach). At the end all of this is validated using experimental datasets (TARGET and in planta). The signal in our case is N, which is Nitrogen because we are in the context of plants treated with nitrogen. Network Walking [16] is a time-based network inference approach which connects rapid and transient TF-targets captured in roots cells, with downstream targets identified in planta. Here, we adapt a time-driven machine learning method we validated [48] [47], to test the hypothesis that rapid and transient direct targets detected in cells, lead to the regulation of downstream

targets in whole plants, using an approach we term Network Walking (A.15A). Our results have shown that Hit-and-Run TFs can effect target gene expression through two distinct mechanisms - Hit-and-Run and stable binding - depending on the cis-context of a target([6]; [41]). Thus, our innovation in Network Walking, is to further adapt the DFG modeling approach to account for the two types of TF targets- Hit-and-Run and stable - and improve the predictive power of our dynamic GRN. Network Walks connect rapid and transient targets in cells, with downstream targets in plants. We successfully implemented the Network Walking approach, and our results on bZIP1 and NLP7 predict that an important set of transient TF-targets identified only in cells (e.g. TF2s) (A.15, inner ring), regulate downstream targets in plants (outer ring) (A.15B&C). The Network walking approach can be subdivided in three steps. In Step 1 of the Network Walk (A.15A), we identify genes involved in bZIP1-mediated N-signaling as the union of: i) TF targets identified in the cell-based TARGET system (primary and transient targets)([6]) and ii) bZIP1 targets identified by TF perturbation in planta (primary and secondary targets)([46]). In order to make time-based connections between these datasets, we intersect this union with the N-regulated genes in roots from a fine-scale time-course study of N-regulated genes in planta([48]). In Step 2, we infer edges between the early N-response bZIP1 targets identified in cells (i.e. transient target TF2s)([6]), and downstream targets identified in planta (i.e. gene Z)([46]), using the time-series N-response data and the time-based network inference method, Dynamic Factor Graph (DFG)([58]), with our own adaptations (described in Krouk et al.([48]; [47])). Briefly, DFG synthesizes Bayesian and Markovian approaches to learn from time-series transcriptome data and to estimate the quantitative influence of up to k TFs (k is an adjustable parameter

currently set at 10) at time t, on target genes at t+1. It then uses these estimates to predict the expression level of target gene Z, at left-out time-points([48]; [47]) (see Background for more details on the performance of DFG). In further iterations of Step 2, we use our new knowledge to improve the predictive power of network models by incorporating both Hit-and-Run and stable relationships for each TF tested. An example of our modeling strategy for the Hit-and-Run targets to account for potential TF partners that continue transcription initiated by the TF Hit, after the Run, is as follows: i) experimentally identify the Hit-and-Run source-target (S-T) relationships, ii) identify cis-elements associated with each Hit-and-Run target, iii) identify partner TFs (P) that bind to those cis-elements, and iv) create non-linear (quadratic) models that account for source (S) and partner (P) effects on target expression. Algorithmically, the relationships between the source (S), partner (P) and target (T) is a prior relationship (i.e. there would be a hyperedge from S, P to T). Standard regression methods can be used, but instead of having only linear terms (i.e., a sum of weighted TF expressions), there would also be quadratic terms (involving S,P). Because the Hit-and-Run sources (S) may recruit a partner (P) and then later affect a target (T), we also have time-offset interaction terms in which we look at the mRNA level of c1* S(t) * P(t) + c2*S(t-1)*P(t) + c3*S(t-2)*P(t). Thus we can model the effects of the Hit-and-Run source at times t, t-1 and time t-2, on the target at time t. For the stable TF-targets, our network models follow the adapted DFG methods we previously validated([48]). Thus, our dynamic network models account for both stable and transiently bound (Hit-and-Run) TF-target interactions. Step 3. The Network Walk resulting from Steps 1 and 2, connect TF targets in cells, with those identified in plants - and this relationship can be visualized using Cytoscape

(A.15B&C). Our Network Walk (Fig. A.15) shows direct targets of bZIP1 identified in root cells (inner ring, A.15B)([6]), are predicted to regulate downstream targets in planta (outer ring A.15B)([46]), including key genes in N-assimilation (e.g. nitrate reductase, NIA1)([**?**]). Remarkably, all 18 TF2s predicted to mediate downstream bZIP1 responses, are transient bZIP1 targets detected only in cells (inner ring, A.15B)([6]). These transient TF2 targets of bZIP1 include LBD38 and LBD39, previously associated with N-response in planta([74]). Our Network Walk now places them downstream of bZIP1 a transcriptional cascade that mediates N-signaling e.g. N->bZIP1->LBD38/39->NIA1 (nitrate reductase). This LBD38/39->NIA1 link is supported by in planta perturbation studies. In our Network Walk for NLP7, the direct targets identified in cells (inner ring Fig. A.15C) are predicted to regulate genes in planta([19]) (outer ring A.15C), defining a Network Walk: N->NLP7->LBD38->NIA1 (nitrate reductase), controlling the first step of N assimilation.

In summary, charting a temporal path in gene networks requires linking early transcription factor (TF)-triggered events to downstream effects(Fig. A.16) [16]. Here, we scale-up a cell-based TF-perturbation assay to identify direct regulated targets of 33 nitrogen (N)-early response TFs encompassing 88% of N-responsive Arabidopsis genes(Fig. A.17)[16]. We uncover a duality where each TF is an inducer and repressor, and in vitro cis-motifs are typically specific to regulation directionality. Validated TF-targets are used to refine precision of a time-inferred root network, connecting 145 N-responsive TFs and 311 targets. To infer the root gene regulatory network, we applied a time-based machine learning method to the

dynamic N-responsive genes of a time-series transcriptome in whole roots. These data are used to chart network paths from direct TF_1-regulated targets identified in cells to indirect targets responding only *in planta* via Network Walking. We uncover network paths from TGA1 and CRF4 to direct TF_2 targets, which in turn regulate 76% and 87% of TF_1 indirect targets *in planta*, respectively(Fig. A.18). These results have implications for N-use and the approach can reveal temporal networks for any biological system [16]. Code for the computational pipeline is available at: github.com/jacirrone/MLTimeSeriesPNASNatComm.

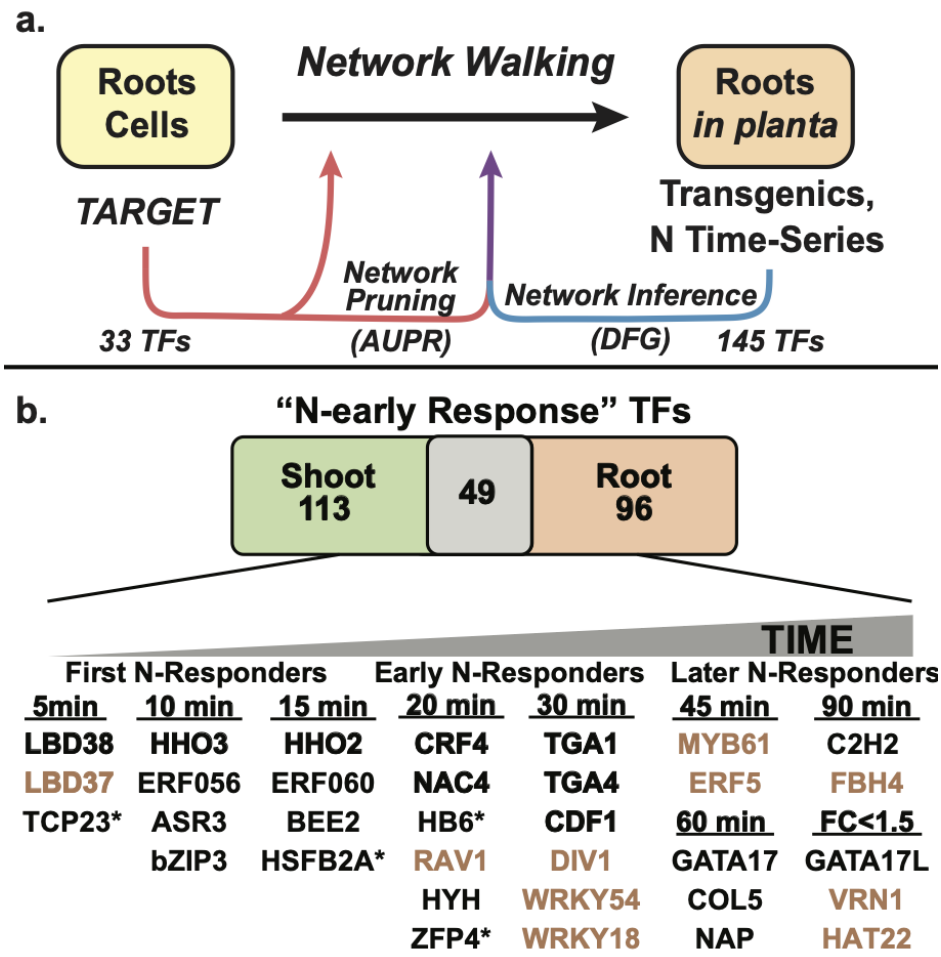


Figure A.16: Network Walking connects validated direct transcription factor (TF) targets to in planta responses[16]. **a** Schematic overview: the Network Walking approach charts a network path from direct targets of a TF identified in cells to its indirect targets, which only respond in planta. This is achieved using data for 33 TF perturbations in root cells using TARGET (Transient Assay Reporting Genome-wide Effects of Transcription factors) scaled-up in this study, and a time-series transcriptome of nitrogen (N) response in whole roots. TF-target edges for 145 TFs were inferred using this time-series data in a machine-learning method called dynamic factor graphs (DFG) (blue arrow). The validated edges and high-confidence inferred edges are used to link a TF to its indirect targets in planta via the Network Walk. **b** The 33 TFs were selected based on their response to N in shoots and roots (black TFs) or roots only (orange TFs) from the N-treatment time-series data of [86]

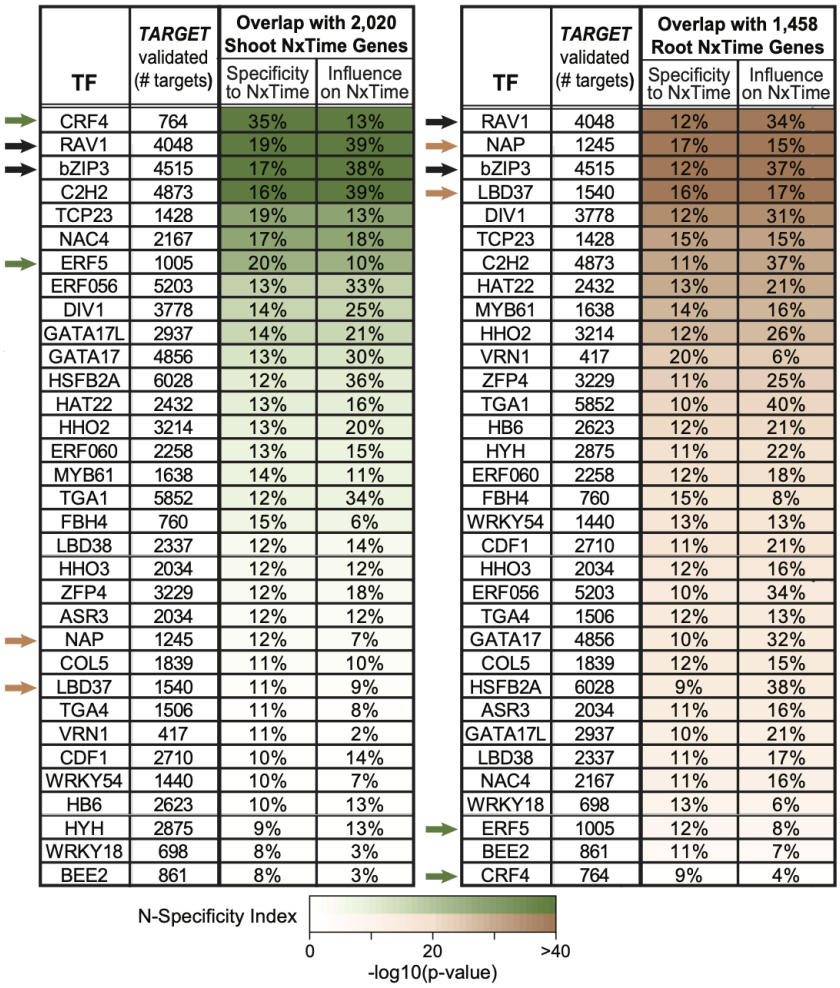


Figure A.17: Validated direct targets of the 33 N-early response TFs are enriched in NxTime genes[16]. The intersection of direct regulated targets for the 33 N-early response TFs identified in root cells using the TARGET system with NxTime genes from [86]. This allowed identification of TFs regulating a significant portion of the N response in both roots and shoots (e.g. bZIP3/RAV1, black arrows). The direct regulated targets of other TFs are enriched in organ-specific NxTime response genes. These include CRF4/ERF5, which are specifically enriched for the shoot NxTime response genes (green arrows), and NAP/LBD37, which are specifically enriched for the root NxTime response genes (orange arrows). Green and orange shading represents the N-Specificity Index29, the p-value calculated using the one proportion z-test.

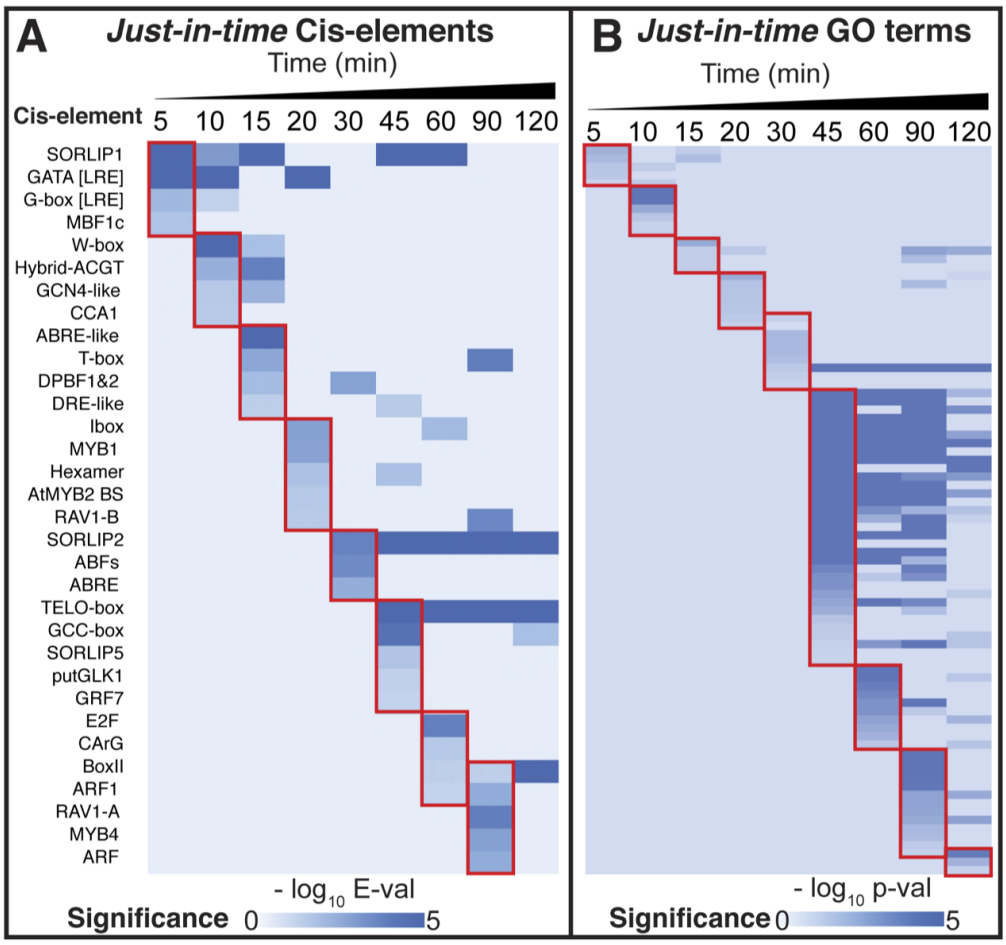


Figure A.4: Genes responding to NxTime by Cubic-Spline Analysis ([55]) were binned into the first time-point at which mean expression changes by >=1.5 fold. A. A cascade of unique cis-element motifs are significantly enriched in each JIT gene set. B. The JIT gene sets have non-overlapping sets of GO-terms enriched at each time-point (Fig. A.2D).

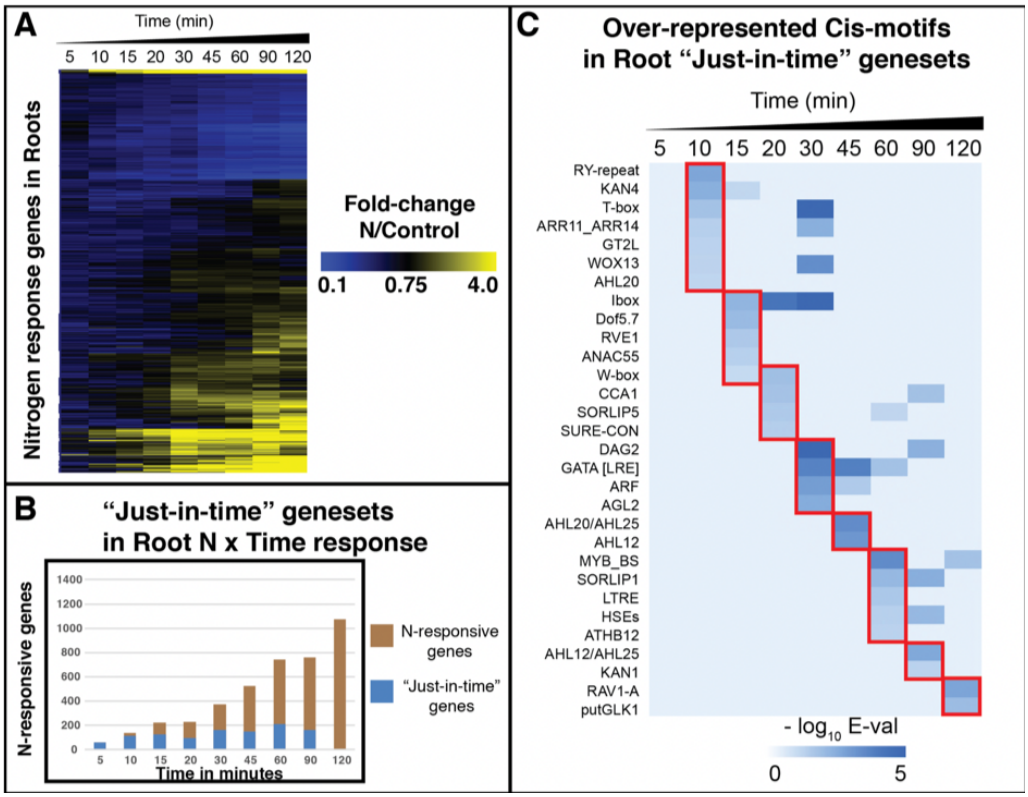


Figure A.5: **Cis-element motif enrichment in just-in-time bins in the root N-response.** A. The transcriptional cascade triggered by N-signal perception shows a sequential activation and repression of 2,468 genes in roots. B. The transcriptional response to N-signal increases over time in roots (brown bars). Just-in-time gene sets (blue bars) are identified using a classification algorithm to capture cohorts of genes whose expression is altered by the N-signal for the first time at that specific time-point C. The set of cis-element motifs specifically enriched in just-in-time analysis of the root NxTime series data is shown. Although, some cis-motifs are shared with the shoot dataset (Fig. A.4A), many of the cis-element motifs in the root just-in-time gene sets are unique to the root N-response (e.g., WOX13, Dof5.7 etc). This result implies that distinct sets of TFs are likely driving the dynamics of the N-signal response in the roots vs. the shoots.

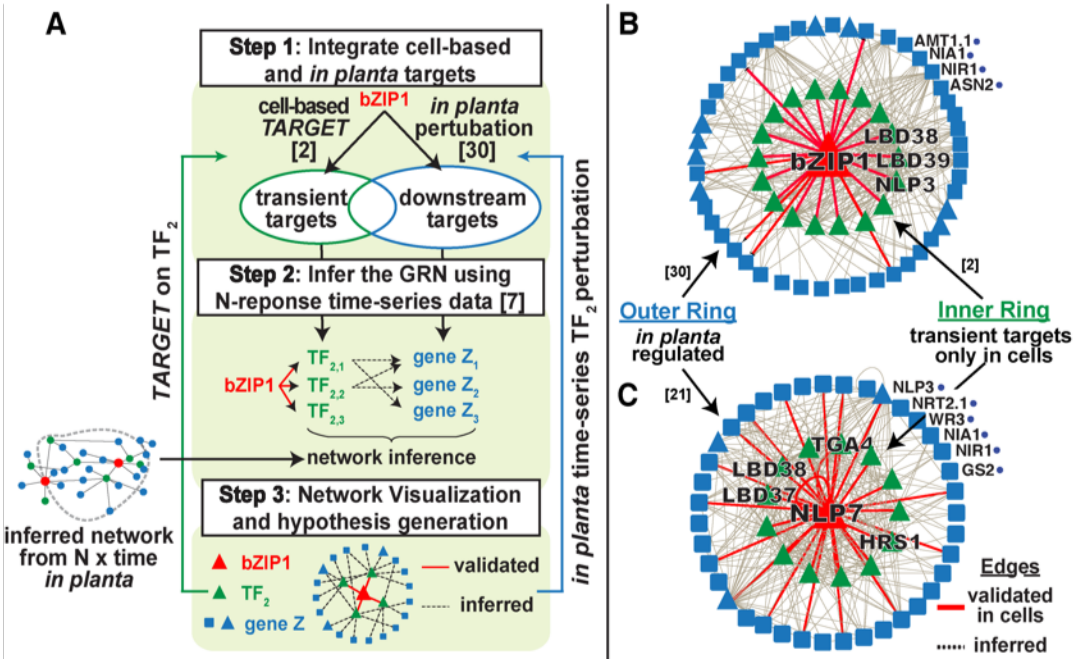


Figure A.15: Network Walking connects transient TF-targets detected cells with downstream responses in planta. A) The Network Walking pipeline. Step 1, overlaps TF-targets detected in cells (Aim 1) with the in planta TF-targets (Aim 2A). Step 2, infers edges between cell and in planta using N-treatment time-series transcriptomic data using the DFG time-based network inference approach. Step 3, networks are visualized using Cytoscape for B) bZIP1 and [C) NLP7.] The transient targets detected in cells (inner ring), are predicted to regulate targets in planta (outer ring), and several examples are validated (e.g. LBD38/39->NIA1)70.

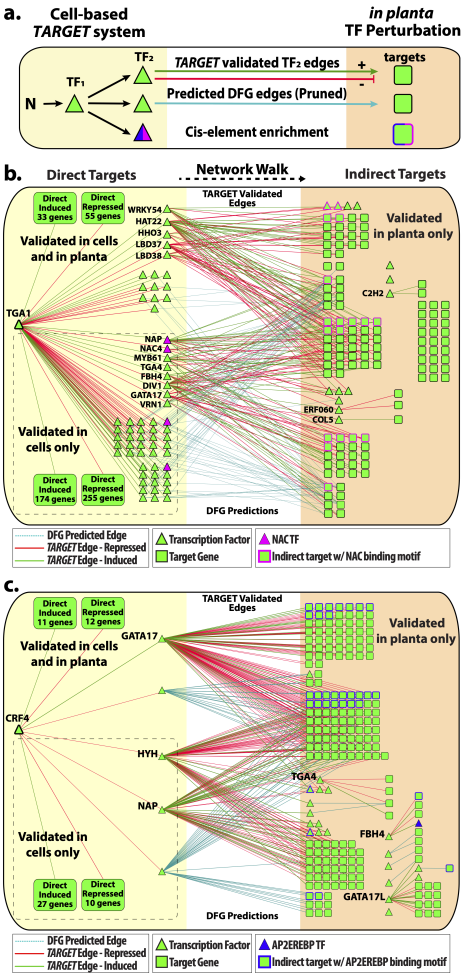


Figure A.18: Network Walking charts a path from direct to indirect TF-targets[16]. (a) A schematic representation of the Network Walking approach used to connect direct TF targets identified in cells, to the indirect targets identified only in planta. Example of Network Walks from direct targets identified in cells (yellow shaded region), to indirect targets identified only in planta (orange shaded region) for (b) TGA1 and (c) CRF4. Edges connecting the indirect targets back to TGA1/CRF4 through their direct TF2 targets come from validated TARGET edges as well as from high-confidence edges from the pruned time-inferred DFG network. Enrichment of the consensus cis-motif for the 80 clusters in the 500bp promoters and gene body of the indirect targets of TGA1 and CRF4 was assessed. The most significant cluster CCM in indirect targets of TGA1 was for cis-motif cluster 15 (NAC family) in the gene body. For CRF4, the CCM for cluster 8 (AP2EREBP) was enriched in the gene body of CRF4 indirect targets. The network shown is limited to TFs and targets that respond to NxTime in [86]. For clarity, edges to target genes include only the top three validated edges based on fold change, and top ten predicted DFG edges based on edge score.

Appendix B

OutPredict in Action

B.1 OutPredict Installation and Run

To run OutPredict, the latest version of Miniconda or Anaconda must be previously installed (Anaconda: https://www.anaconda.com/distribution/#download-section).

If conda is already installed on your machine, you can update to the latest version with:

`conda update – all`

Clone the codebase:

`git clone https : //github.com/jacirrone/OutPredict.git`

Enter the `OP_3` directory:

`cd OP_3/`

To install OutPredict, first install the OpenMP library as follows:

For Mac-OS:

`brew install libomp`

For Linux:

(Side note: it might be necessary to run "sudo apt-get update" and "sudo apt-get install gcc", especially if you are using a virgin AWS machine, for example)

`sudo apt − get install libomp − dev clang`

Then, run (in the `OP3/directory`) the OutPredict Installation file:

`sh install.sh`

As example to run OutPredict, invoke the corresponding pipeline script for the dream10 dataset:

`python dream10_pipeline.py`

B.1.1 Required data for OutPredict

The Datasets directory, "`OP_3/Datasets/`", contains the directories of each organism's datasets.

Let us consider a dataset for a generic organism called "`new_organism`".

Inside this directory "`OP_3/Datasets/new_organism/`" the following sample files are required (for the "`dream10`" example the directory is "`OP_3/Datasets/dream10/`") :

B.1.2 expression.tsv

Expression values; must include row (genes) and column (conditions) names

Obtain expression data and save it as a tsv file "`expression.tsv`" of [`Genes x Samples`]

B.1.3 gold_standard.tsv

This file is required when choosing the gold standard priors option on OutPredict, see A.3 section.

Needed for OutPredict with "gold_standard" priors; matrix of 0s and 1s indicating whether we have prior knowledge about the interaction of a transcription factor (TF) and a gene; one row for each gene, one column for each TF; must include row (genes) and column (TF) names (Figure B.1).

So the position tf t and gene g: is 1 if there is an inductive or repressive edge; is 0 if there is no such edge or unknown.

Obtain gold standard data, interactions between TFs and target genes and save it as a tsv file "`gold_standard.tsv`" [`Genes x TFs`]

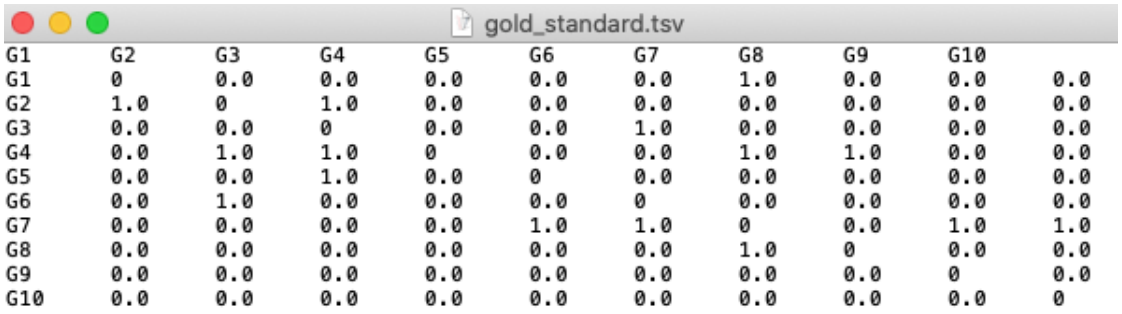


Figure B.1: Gold Standard file example

B.1.4 interaction_weights_list.tsv

This file is required when choosing the steady state priors option on OutPredict, see A.3 example.

Needed for OutPredict with "`steady_state`" priors;

This type of priors is represented by a list of interactions indicating whether we have prior knowledge about the interaction of a transcription factor (TF) and a gene. In this case the prior knowledge is represented by a real number weight, which is an interaction confidence score.

First column are TFs, second column are genes, third column are real number weights.

B.1.5 meta_data.tsv

In a gene expression dataset a condition is defined as an experimental assay or a replicate of an experiment.

The meta data file describes the conditions; must include column names; has five columns (Figure B.2):

`isTs` : `TRUE` if the condition is part of a time-series, FALSE else.

`is1stLast`: "e" if the condition is not part of a time-series; "f" if first; "m" middle; "l" last. Thus "l" means a value at the last time point, "f" at the first time point, "m" all others.

`prevCol`: name of the preceding condition in time-series; NA if "e" or "f".

`del.t`: time in whatever the common unit (e.g. typically minutes for transcription factors) since prevCol; NA if "e" or "f".

`condName`: name of the condition.

meta_data.tsv

```
"isTs"  "is1stLast"     "prevCol"       "del.t" "condName"
FALSE   "e"     NA      NA      "G1(-/-)"
FALSE   "e"     NA      NA      "G2(-/-)"
FALSE   "e"     NA      NA      "G3(-/-)"
FALSE   "e"     NA      NA      "G4(-/-)"
FALSE   "e"     NA      NA      "G5(-/-)"
FALSE   "e"     NA      NA      "G6(-/-)"
FALSE   "e"     NA      NA      "G7(-/-)"
FALSE   "e"     NA      NA      "G8(-/-)"
FALSE   "e"     NA      NA      "G9(-/-)"
FALSE   "e"     NA      NA      "G10(-/-)"
FALSE   "e"     NA      NA      "wt0"
FALSE   "e"     NA      NA      "wt1"
FALSE   "e"     NA      NA      "wt2"
FALSE   "e"     NA      NA      "wt3"
FALSE   "e"     NA      NA      "wt4"
FALSE   "e"     NA      NA      "wt5"
FALSE   "e"     NA      NA      "wt6"
FALSE   "e"     NA      NA      "wt7"
FALSE   "e"     NA      NA      "wt8"
FALSE   "e"     NA      NA      "wt9"
FALSE   "e"     NA      NA      "wt10"
FALSE   "e"     NA      NA      "wt11"
FALSE   "e"     NA      NA      "wt12"
FALSE   "e"     NA      NA      "wt13"
FALSE   "e"     NA      NA      "wt14"
FALSE   "e"     NA      NA      "wt15"
FALSE   "e"     NA      NA      "wt16"
FALSE   "e"     NA      NA      "wt17"
FALSE   "e"     NA      NA      "wt18"
FALSE   "e"     NA      NA      "wt19"
TRUE    "f"     NA      NA      "TS_1delt_0"
TRUE    "m"     "TS_1delt_0"    50      "TS_1delt_50"
TRUE    "m"     "TS_1delt_50"   50      "TS_1delt_100"
TRUE    "m"     "TS_1delt_100"  50      "TS_1delt_150"
TRUE    "m"     "TS_1delt_150"  50      "TS_1delt_200"
TRUE    "m"     "TS_1delt_200"  50      "TS_1delt_250"
TRUE    "m"     "TS_1delt_250"  50      "TS_1delt_300"
TRUE    "m"     "TS_1delt_300"  50      "TS_1delt_350"
TRUE    "m"     "TS_1delt_350"  50      "TS_1delt_400"
TRUE    "m"     "TS_1delt_400"  50      "TS_1delt_450"
TRUE    "l"     "TS_1delt_450"  50      "TS_1delt_500"
TRUE    "f"     NA      NA      "TS_2delt_500"
TRUE    "m"     "TS_2delt_500"  50      "TS_2delt_550"
TRUE    "m"     "TS_2delt_550"  50      "TS_2delt_600"
TRUE    "m"     "TS_2delt_600"  50      "TS_2delt_650"
TRUE    "m"     "TS_2delt_650"  50      "TS_2delt_700"
TRUE    "m"     "TS_2delt_700"  50      "TS_2delt_750"
TRUE    "m"     "TS_2delt_750"  50      "TS_2delt_800"
TRUE    "m"     "TS_2delt_800"  50      "TS_2delt_850"
TRUE    "m"     "TS_2delt_850"  50      "TS_2delt_900"
TRUE    "m"     "TS_2delt_900"  50      "TS_2delt_950"
TRUE    "l"     "TS_2delt_950"  50      "TS_2delt_1000"
TRUE    "f"     NA      NA      "TS_3delt_0"
TRUE    "m"     "TS_3delt_0"    50      "TS_3delt_50"
TRUE    "m"     "TS_3delt_50"   50      "TS_3delt_100"
TRUE    "m"     "TS_3delt_100"  50      "TS_3delt_150"
TRUE    "m"     "TS_3delt_150"  50      "TS_3delt_200"
TRUE    "m"     "TS_3delt_200"  50      "TS_3delt_250"
TRUE    "m"     "TS_3delt_250"  50      "TS_3delt_300"
TRUE    "m"     "TS_3delt_300"  50      "TS_3delt_350"
TRUE    "m"     "TS_3delt_350"  50      "TS_3delt_400"
TRUE    "m"     "TS_3delt_400"  50      "TS_3delt_450"
TRUE    "l"     "TS_3delt_450"  50      "TS_3delt_500"
TRUE    "f"     NA      NA      "TS_4delt_500"
TRUE    "m"     "TS_4delt_500"  50      "TS_4delt_550"
TRUE    "m"     "TS_4delt_550"  50      "TS_4delt_600"
TRUE    "m"     "TS_4delt_600"  50      "TS_4delt_650"
TRUE    "m"     "TS_4delt_650"  50      "TS_4delt_700"
TRUE    "m"     "TS_4delt_700"  50      "TS_4delt_750"
TRUE    "m"     "TS_4delt_750"  50      "TS_4delt_800"
TRUE    "m"     "TS_4delt_800"  50      "TS_4delt_850"
TRUE    "m"     "TS_4delt_850"  50      "TS_4delt_900"
TRUE    "m"     "TS_4delt_900"  50      "TS_4delt_950"
TRUE    "l"     "TS_4delt_950"  50      "TS_4delt_1000"
```

Figure B.2: Meta-data file example.

B.1.6 tf_names.tsv

One TF (transcription factor) name on each line; these must be subset of the row names of the expression data

Create a list of TFs to model for inference and save it as a file "`tf_names.tsv`" with each TF on a separate line [TFs]

Note that each gene (TFs and others) must have the same name in all files (expression, gold_standard, etc.)

B.1.7 Construct a new run script for a generic organism

Here is an example of the contents of that file (‘pipeline_new_organism.py‘):

```
1
2  if __name__ == '__main__':
3
4      #Create an OutPredict instance
5
6      op = OutPredict()
7
8      #Set required file names and parameters:
9
10     op.num_of_trees = 300  # The number of Trees
11     #for Random Forests
12
13     op.input_dir_name = "dream10"  # Name of
14     #Directory, inside OP_3/Datasets/, containing the dataset
15
```

```
op.test_set_split_ratio = 0.15  # The percentage of
#data points to use for the test set separately for
#time-series and steady-state, e.g. 0.15, 15\% of
#steady-state data will be used as test set,
#15\% of the time-series data (last time points of time-series)

op.training_data_type = "TS-SS"  # whether to use for
#training TS(time-series), SS(steady-stae) or TS-SS
#(time-series and steady-state)

op.leave_out_data_type = "TS"  # whether to use for
#training TS(time-series), SS(steady-stae) or TS-SS
#(time-series and steady-state)

op.genes_coeff_of_var_threshold = 0  # coefficient of
#variance threshold to filter the genes to modeling;
#0 to modeling all genes

op.num_of_cores = 20  # (Integer) number of
#cores to use for parallelization

#it's not necessary to set which method to use -
#either time-step or ode-log - because
#it will be automatically learned

#Set required params to run OutPredict WITH Priors:
```

```
42
43      op.prior_file_name = "gold_standard.tsv" #either
44      #name of file containing prior knowledge or empty
45
46      op.priors = "gold_standard" #"steady_state"
47      # gold_standard or steady_state or empty
48
49      op.run()
```

B.1.8 Run OutPredict

Enter the OP_3 directory and activate the conda environment op3

`cd` OP_3/

`conda activate op3`

To use OutPredict WITHOUT priors, do NOT set the params "prior_file_name" and "priors", and this script can now be run from the command line as (Figure B.3):

```
1 python -s pipeline\_new\_organism.py
```

To use OutPredict WITH priors, after properly setting BOTH the params "prior_file_name" and "priors", this script can now be run from the command line as (Figure B.4):

```
1 python pipeline\_new\_organism.py
```

The folder "OP_3/output/" contains the output folders for the different runs of OutPredict.

A generic output folder for a run related to the "WITHOUT priors" version is called "new_organism_output_RF_...".

```
:::::::RUNNING OUTPREDICT WITHOUT PRIORS:::::::::
Loading and Preprocessing Dataset  dream10 ...

Write output to file _summary_results.txt inside the output folder

Learning the OutPredict Model ...

OutPredict has found the best model according to the out-of-bag score...

The best model is compared to the Penultimate Value Naive approach.

MSE Pen. Value Naive Time-Series Test set: 0.0067437605855904
MSE Time-Series Test Set of the best model is: 0.005308657531077606

 Influences Inference: Causal connections from transcription factors to genes are
 printed to file as a ranked list of interactions (Ranked_list_TF_gene) and as a
matrix (Matrix_TF_gene) inside the output folder
```

Figure B.3: Running OutPredict without Priors

```
(op3) Jacopos-MacBook-Pro-2:OP_3 Jacopo$ python pipeline_dream10.py
:::::::RUNNING OUTPREDICT WITH PRIORS:::::::::
Loading and Preprocessing Dataset  dream10 ...

Write output to file _summary_results.txt inside the output folder

Learning the OutPredict Model ...

OutPredict has found the best model according to the out-of-bag score...

The best model is compared to the Penultimate Value Naive approach.

MSE Pen. Value Naive Time-Series Test set: 0.0067437605855904
MSE Time-Series Test Set of the best model is: 0.004963031286149499

 Influences Inference: Causal connections from transcription factors to genes are
 printed to file as a ranked list of interactions (Ranked_list_TF_gene) and as a
matrix (Matrix_TF_gene) inside the output folder
```

Figure B.4: Running OutPredict with Priors

A generic output folder for a run related to the "WITH priors" version is called "new_organism_output_RF-mod_..."

Bibliography

[1] C. A et al. The transplanta collection of arabidopsis lines: a resource for functional analysis of transcription factors based on their conditional overexpression. *Plant J*, 77(6):944–953, 2014.

[2] G. A and B. SM. Mapping transcriptional networks in plants: Data- driven discovery of novel biological mechanisms. *Annu Rev Plant Biol*, 67:575–594, 2016.

[3] K. A. Plant nitrogen assimilation and its regulation: a complex puzzle with missing pieces. *Curr Opin Plant Biol*, 25:115–122, 2015.

[4] M. A et al. Atnigt 1/hrs1 integrates nitrate and phosphate signals at the arabidopsis root tip. *Nat Commun*, 6:6274, 2015.

[5] M. A and K. G. The primary nitrate response: a multifaceted signalling pathway. *J Exp Bot*, 65(19):5567–5576, 2014.

[6] P. A et al. Hit-and-run transcriptional control by bzip1 mediates rapid nutrient signaling in arabidopsis. *Proc Natl Acad Sci U S A*, 111(28):10371–10376, 2014.

[7] Y. A et al. *AGRIS: the Arabidopsis Gene Regulatory Information Server*. an update, 2011.

[8] B. AJ. Photorespiration and nitrate assimilation: a major intersection between plant carbon and nitrogen. *Photosynth Res*, 123(2):117–128, 2015.

[9] S. AM et al. Mapping and dynamics of regulatory DNA and transcription factor networks in A. *thaliana. Cell Rep*, 8(6):2015–2030, 2014.

[10] M. L. Arrieta-Ortiz, C. Hafemeister, A. R. Bate, T. Chu, A. Greenfield, B. Shuster, S. N. Barry, M. Gallitto, B. Liu, T. Kacmarczyk, F. Santoriello, J. Chen, C. D. Rodrigues, T. Sato, D. Z. Rudner, A. Driks, R. Bonneau, and P. Eichenberger. An experimentally supported model of the bacillus subtilis global transcriptional regulatory network. *Molecular System Biology*, 2015.

[11] E. Bastakis, B. Hedtke, C. Klermund, B. Grimm, and C. Schwechheimer. Llm-domain b-gata transcription factors play multifaceted roles in controlling greening in arabidopsis. *Plant Cell*, 2018.

[12] C. Behringer, E. Bastakis, Q. Ranftl, K. Mayer, and C. Schwechheimer. Functional diversification within the family of b-gata transcription factors through the leucine-leucine-methionine domain. *Plant Physiology*, 2014.

[13] B. BO et al. Target: a transient transformation system for genome-wide transcription factor target discovery. *Mol Plant*, 6(3):978–980, 2013.

[14] L. Breiman. Classification and regression trees. *Chapman & Hall CRC*, 1984.

[15] L. Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science 16 (3). Institute of Mathematical Statistics: 199-23*, 2001.

[16] M. D. Brooks, J. Cirrone, A. Pasquino, J. M. Alvarez, J. Swift, S. Mittal, C.-L. Juang, K. Varala, R. A. Gutierrez, G. Krouk, D. Shasha, , and G. M. Coruzzi. Network walking charts transcriptional pathways for dynamic nitrogen signaling using validated and predicted genome-wide interactions. *Nature Communication*, 2019.

[17] R. Bustos, G. Castrillo, F. Linhares, M. I. Puga, V. Rubio, J. Pérez-Pérez, R. Solano, A. Leyva, and J. Paz-Ares. A central regulatory system largely controls transcriptional activation and repression responses to phosphate starvation in arabidopsis. *Plos Genetics*, 2010.

[18] P. CA, B.-W. V, D. KJ, and W. DL. *Nonparametric Bayesian inference for perturbed and orthologous gene regulatory networks.* Bioinformatics 28(12):i233-241, 2012.

[19] e. a. Castaings L. The nodule inception-like protein 7 modulates nitrate sensing and metabolism in arabidopsis. *Plant J*, 2009.

[20] L. E. Chai, S. K. Loh, S. T. Low, M. S. Mohamad, S. Deris, and Z. Zakaria. A review on the computational approaches for gene regulatory network construction. *Computers in Biology and Medicine, 48, 55-65*, 2014.

[21] P. Christopher and W. David. How to infer gene networks from expression profiles. *Interface Focus*, 2011.

[22] J. Cirrone, M. D. Brooks, R. Bonneau, G. M. Coruzzi, and D. E. Shasha. Outpredict: multiple datasets can improve prediction of expression and inference of causality. *Scientific Reports*, 2020.

[23] B. CM. *Pattern recognition and machine learning (Springer, New York) pp xx.* 738, 2006.

[24] G. Coruzzi, K. Varala, A. Marshall-Colon, M. Brooks, S. Ruffel, J. Alvarez, A. Pasquino, J. Cirrone, and D. Shasha. The 4th dimension of transcriptional networks: Time. *The FASEB Journal*, 2019.

[25] P. CS et al. Sungear: interactive visualization and functional analysis of genomic datasets. *Bioinformatics*, 23(2):259–261, 2007.

[26] Y. D et al. Nin-like protein 8 is a master regulator of nitrate-promoted seed germination in Arabidopsis. *Nat Commun*, 7:13179, 2016.

[27] F. M. Delgado and F. GAmez-Vela. Computational methods for gene regulatory networks reconstruction and analysis: A review. *Artificial Intelligence in Medicine, Volume 95*, 2019.

[28] H. EA, S. R, S. D, and Q. J. Rna-seq analysis in mev. *Bioinformatics*, 27(22):3209–3210, 2011.

[29] V. EA, A. JM, and G. RA. *Nitrate regulation of AFB3 and NAC4 gene expression in Arabidopsis roots depends on NRT1.* 1 nitrate transport function, 2014.

[30] M. Fan, M. Bai, J. Kim, T. Wang, E. Oh, L. Chen, C. Park, S. Son, S. Kim, M. Mudgett, and Z. Wang. The bhlh transcription factor hbi1 mediates the

trade-off between growth and pathogen-associated molecular pattern-triggered immunity in arabidopsis. *Plant Cell*, 2014.

[31] S. G, P. RJ, and C. A. Lessons from the dream2 challenges. *Ann N Y Acad Sci*, 1158:159–195, 2009.

[32] A. Gitter, Z. Siegfried, M. Klutstein, O. Fornes, B. Oliva, I. Simon, and Z. Bar-Joseph. Backup in gene regulatory networks explains differences between binding and knockout results. *Molecular System Biology*, 2009.

[33] A. Greenfield, C. Hafemeister, and R. Bonneau. Robust data-driven incorporation of prior knowledge into the inference of dynamic regulatory networks. *Bioinformatics*, 2013.

[34] A. Greenfield, A. Madar, H. Ostrer, and R. Bonneau. Dream4: Combining genetic and dynamic information to identify biological networks and dynamical models). *Edited by Mark Isalan. PLoS ONE 5 (10). Public Library of Science (PLoS): e13397*, 2010.

[35] V. Gregis, F. Andrés, A. Sessa, R. Guerra, S. Simonini, J. Mateos, S. Torti, F. Zambelli, G. Prazzoli, K. Bjerkan, P. Grini, G. Pavesi, L. Colombo, G. Coupland, and M. Kater. Identification of pathways directly regulated by short vegetative phase during vegetative and reproductive development in arabidopsis. *Genome Biology*, 2013.

[36] W. H. *Ggplot2 : elegant graphics for data analysis (Springer, New York) pp viii.* 212, 2009.

[37] S. D. Hooper, S. Boue, R. Krause, L. Jensen, C. Mason, M. Ghanim, K. White, E. Furlong, and P. Bork. Identification of tightly regulated groups of genes

during drosophila melanogaster embryogenesis. *Molecular System Biology*, 2007.

[38] V. A. Huynh-Thu and P. Geurts. Dyngenie3: dynamical genie3 for the inference of gene networks from time series expression data. *Scientific Reports*, 2018.

[39] V. A. Huynh-Thu, A. Irrthum, L. Wehenkel, and P. Geurts. Inferring regulatory networks from expression data using tree-based methods. *Edited by Mark Isalan. PLoS ONE 5 (9). Public Library of Science (PLoS): e12776*, 2010.

[40] C. J, M. TC, V. E, and G. RA. Systems analysis of transcriptome data provides new hypotheses about Arabidopsis root response to nitrate treatments. *Front Plant Sci*, 5:22, 2014.

[41] D. J et al. Hit-and-run transcription: de novo transcription initiated by a transient bzip1 hit persists after the run. *BMC Genomics*, 17:92, 2016.

[42] A. JM et al. Systems approach identifies TGA1 and TGA4 transcription factors as important regulatory components of the nitrate response of Arabidopsis thaliana roots. *Plant J*, 80(1):1–13, 2014.

[43] S. Jozefczuk, S. Klie, G. Catchpole, J. Szymanski, A. Cuadros-Inostroza, D. Steinhauser, J. Selbig, and L. Willmitzer. Metabolomic and transcriptomic stress response of escherichia coli. *Molecular System Biology*, 2010.

[44] E. JR. *Base-pair Resolution Atlases of the Plant Cistrome and Epicistrome*. 2017.

[45] P. K et al. Distinct signalling pathways and transcriptome response signatures differentiate ammonium- and nitrate-supplied plants. *Plant Cell Environ*, 33(9):1486–1501, 2010.

[46] S. G. Kang, J. Price, P. C. Lin, J. C. Hong, and J. C. Jang. The arabidopsis bzip1 transcription factor is involved in sugar signaling, protein networking, and dna binding. *Molecular Plant*, 2010.

[47] G. Krouk, J. Lingeman, A. M. Colon, G. Coruzzi, and D. Shasha. Gene regulatory networks in plants: learning causality from time and perturbation. *Genome Biol, 14: 123*, 2013.

[48] G. Krouk, P. Mirowski, Y. LeCun, D. Shasha, and G. Coruzzi. Predictive network modeling of the high-resolution dynamic plant transcriptome in response to nitrate. *Genome Biol, 11: R123*, 2010.

[49] X. Luo, W. Lin, S. Zhu, J. Zhu, Y. Sun, X. Fan, M. Cheng, Y. Hao, E. Oh, M. Tian, L. Liu, M. Zhang, Q. Xie, K. Chong, and Z. Wang. Integration of light-and-brassinosteroid signaling pathways by a gata transcription factor in arabidopsis. *Developmental Cell*, 2010.

[50] W. M et al. *LegumeGRN: a gene regulatory network prediction server for functional and comparative studies*. PLoS One 8(7):e67434, 2013.

[51] G. Mallikarjuna, K. Mallikarjuna, M. K. Reddy, and T. Kaul. Expression of osdreb2a transcription factor confers enhanced dehydration and salt stress tolerance in rice. *Biotechnol. Lett.*, 2011.

[52] D. Marbach, J. C. Costello, R. Kuffner, N. Vega, R. J. Prill, D. M. Camacho, K. R. Allison, the DREAM5 Consortium, M. Kellis, J. J. Collins, and

G. Stolovitzky. Wisdom of crowds for robust gene network inference. *Nature Methods*, 2012.

[53] C. Marchive, F. Roudier, L. Castaings, V. Bréhaut, E. Blondet, V. Colot, C. Meyer, and A. Krapp. Nuclear retention of the transcription factor nlp7 orchestrates the early response to nitrate in plants. *Nature Communications*, 2013.

[54] M. Maziarz. A review of the granger-causality fallacy. *The Journal of Philosophical Economics: Reflections on Economic and Social Issues. VIII*, 2015.

[55] R. ME et al. *limma powers differential expression analyses for RNA- sequencing and microarray studies.* Nucleic Acids Res 43(7):e47, 2015.

[56] R. Michna, F. Commichau, D. Todter, C. Zschiedrich, and J. Stulke. Subtiwiki-a database for the model organism bacillus subtilis that links pathway, interaction and expression information. *Nucleic Acids Research 42:D692 - D698*, 2014.

[57] P. Mirowski. Time series modeling with hidden variables and gradient-based algorithms. *Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, Ph. D. Dissertation*, 2011.

[58] P. Mirowski and Y. LeCun. Dynamic factor graphs for time series modeling. *Machine Learning and Knowledge Discovery in Databases, Pt Ii, 5782: 128-43*, 2009.

[59] S. Miyashima, P. Roszak, I. Sevilem, K. Toyokura, J. H. B. Blob, N. Mellor, H. Help-Rinta-Rahko, S. Otero, W. Smet, M. Boekschoten, G. Hooiveld,

K. Hashimoto, O. Smetana, R. Siligato, E. Wallner, A. P. Mahonen, Y. Kondo, C. W. Melnyk, T. Greb, K. Nakajima, R. Sozzani, A. Bishopp, B. D. Rybel, and Y. Helariutta. Mobile pear transcription factors integrate positional cues to prime cambial growth. *Nature. 565, 490-494*, 2019.

[60] K. MS et al. Virtualplant: a software platform to support systems biology research. *Plant Physiol*, 152(2):500–515, 2010.

[61] T. Murali, S. Pacifico, J. Yu, S. Guest, G. r. Roberts, and R. J. Finley. Droid 2011: a comprehensive, integrated resource for protein, transcription factor, rna and gene interactions for drosophila. *Nucleic Acids Research*, 2011.

[62] B. N. Feeding a hungry world. *Science*, 318(5849):359, 2007.

[63] P. Nicolas, U. Mader, D. E., T. Rochat, A. Leduc, N. Pigeonneau, E. Bidnenko, E. Marchadier, M. Hoebeke, S. Aymerich, D. Becher, P. Bisicchia, E. Botella, O. Delumeau, G. Doherty, E. Denham, M. Fogg, V. Fromion, A. Goelzer, A. Hansen, E. Hartig, C. Harwood, G. Homuth, H. Jarmer, M. Jules, E. Klipp, L. Le Chat, F. Lecointe, P. Lewis, W. Liebermeister, A. March, R. Mars, P. Nannapaneni, D. Noone, S. Pohl, B. Rinn, F. Rugheimer, P. Sappa, F. Samson, M. Schaffer, B. Schwikowski, L. Steil, J. Stulke, T. Wiegert, K. Devine, A. Wilkinson, J. Van Dijl, M. Hecker, U. Volker, P. Bessieres, and P. Noirot. Condition-dependent transcriptome reveals high-level regulatory architecture in bacillus subtilis. *Science*, 2012.

[64] N. L. Novere. Quantitative and logic modelling of molecular and gene networks. *Nature Reviews Genetetics, 16, 146-158*, 2015.

[65] W. O et al. Egrins (environmental gene regulatory influence networks) in rice that function in the response to water deficit, high temperature, and agricultural environments. *Plant Cell*, 28(10):2365–2384, 2016.

[66] G. P et al. Interacting tcp and nlp transcription factors control plant responses to nitrate availability. *Proc Natl Acad Sci U S A*, 114(9):2419–2424, 2017.

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

[68] F. Petralia, P. Wang, J. Yang, , and Z. Tu. Integrative random forest for gene regulatory network inference). *Bioinformatics 31 (12). Oxford University Press (OUP)*, 2015.

[69] Z. PJ, C. MA, A. CI, and R. AM. Cytokinin response factor 4 (CRF4) is induced by cold and involved in freezing tolerance. *Plant Cell Rep*, 35(3):573–584, 2016.

[70] W. R et al. Genomic analysis of the nitrate response using a nitrate reductase-null mutant of Arabidopsis. *Plant Physiol*, 136(1):2512–2522, 2004.

[71] W. R, O. M, X. X, and C. NM. Microarray analysis of the nitrate response in Arabidopsis roots and shoots reveals over 1, 000 rapidly responding genes and new linkages to glucose, trehalose-6-phosphate, iron, and sulfate metabolism. *Plant Physiol*, 132(2):556–567, 2003.

[72] G. RA et al. Systems approach identifies an organic nitrogen-responsive gene network that is regulated by the master clock control gene CCA1. *Proc Natl Acad Sci U S A*, 105(12):4939–4944, 2008.

[73] O. RC et al. Cistrome and epicistrome features shape the regulatory dna landscape. *Cell*, 166(6):1598, 2016.

[74] G. Rubin, T. Tohge, F. Matsuda, K. Saito, and W.-R. Scheible. Members of the lbd family of transcription factors repress anthocyanin synthesis and affect additional nitrogen responses in arabidopsis. *Plant Cell*, 2009.

[75] F. S et al. An arabidopsis t-dna mutant affected in nrt2 genes is impaired in nitrate uptake. *FEBS Lett*, 2001.

[76] M. S et al. Transcript profiling in the chl1-5 mutant of arabidopsis reveals a role of the nitrate transporter nrt1.1 in the regulation of another nitrate transporter, nrt2.1. *Plant Cell*, 16(9):2433–2447, 2004.

[77] R. S, P. A, K. G, C. GM, and L. B. Long-distance nitrate signaling displays cytokinin dependent and independent branches. *J Integr Plant Biol*, 58(3):226–229, 2016.

[78] R. S et al. Nitrogen economics of root foraging: transitive closure of the nitrate-cytokinin relay and distinct systemic signaling for N supply vs demand. *Proc Natl Acad Sci U S A*, 108(45):18524–18529, 2011.

[79] H. Salgado, M. Peralta-Gil, S. Gama-Castro, A. Santos-Zavaleta, L. Muniz-Rascado, J. S. Garcıa-Sotelo, V. Weiss, H. Solano-Lira, I. Martınez-Flores, A. Medina-Rivera, G. Salgado-Osorio, S. Alquicira-Hernandez, K. Alquicira-

Hernandez, A. Lopez-Fuentes, L. Porron-Sotelo, A. M. Huerta1, C. Bonavides-Martınez, Y. Balderas-Martınez, L. Pannier, M. Olvera, A. Labastida, V. Jimenez-Jacinto, L. Vega-Alvarado, V. del Moral-Chavez, A. Hernandez-Alvarez, E. Morett, and J. Collado-Vides. Regulondb v8.0: omics data sets, evolutionary conservation, regulatory phrases, cross-validated gold standards and more. *Nucleic Acids Research 41, D203-D213*, 2013.

[80] T. Schaffter, D. Marbach, and D. Floreano. Genenetweaver: In silico benchmark generation and performance profiling of network inference methods. *Bioinformatics, 27(16):2263-70*, 2011.

[81] M. Slattery, T. Zhou, L. Yang, A. C. D. Machado, R. Gordan, and R. Rohs. Absence of a simple code: how transcription factors read the genome. *Trends in Biochemical Sciences; 39(9): 381-399*, 2014.

[82] M. R. Smith, M. Clement, T. Martinez, and Q. Snell. Time series gene expression prediction using neural networks with hidden layers. *BIOT*, 2010.

[83] J. Swift, M. Adame, D. Tranchina, A. Henry, and G. M. Coruzzi. Water impacts nutrient dose responses genome-wide to affect crop production. *Nature Communications*, 2019.

[84] M. TC et al. The diurnal project: Diurnal and circadian expression profiling, model-based pattern matching, and promoter analysis. *Cold Spring Harb Symp Quant Biol*, 72:353–363, 2007.

[85] B. U et al. Time-series transcriptomics reveals that agamous-like22 affects primary metabolism and developmental processes in drought-stressed arabidopsis. *Plant Cell*, 28(2):345–366, 2016.

[86] K. Varala, A. Marshall-Colon, J. Cirrone, M. D. Brooks, A. V. Pasquino, S. Léran, S. Mittal, T. M. Rock, M. B. Edwards, G. J. Kim, S. Ruffel, W. R. McCombie, D. Shasha, , and G. M. Coruzzi. Temporal transcriptional logic of dynamic regulatory networks underlying nitrogen signaling and use in plants. *Proceedings of the National Academy of Sciences(PNAS)*, 2018.

[87] L. W and B. C. Pathview: an r/bioconductor package for pathway- based data integration and visualization. *Bioinformatics*, 2013.

[88] Z. X et al. Managing nitrogen for sustainable development. *Nature*, 528(7580):51–59, 2015.

[89] L. Y, S. K, and H. ME. Rapid. *organ-specific transcriptional responses to light regulate photomorphogenic development in dicot seedlings.*, 156(4):2124–2140, 2011.

[90] L. Y, P. SA, and J. SA. Gene networks in plant biology: Approaches in reconstruction and analysis. *Trends Plant Sci*, 20(10):664–675, 2015.

[91] L. Y and von Wiren N. Ammonium as a signal for physiological and morphological responses in plants. *J Exp Bot*, 68(10):2581–2592, 2017.

[92] U. Yamanouchi, M. Yano, H. Lin, A. M., and K. Yamada. A rice spotted leaf gene, spl7, encodes a heat stress transcription factor protein. *Proc. Natl. Acad. Sci.*, 2002.

[93] W. YY, H. PK, and T. YF. Uptake, allocation and signaling of nitrate. *Trends Plant Sci*, 17(8):458–467, 2012.

[94] C. Zou and J. Feng. Granger causality vs. dynamic bayesian network inference: a comparative study. *BMC Bioinformatics*, 2009.