OutPredict: multiple datasets can improve prediction of expression and inference of causality

Jacopo Cirrone[1,*], **Matthew D. Brooks**[2], **Richard Bonneau**[1,2,3], **Gloria M. Coruzzi**[2], **and Dennis E. Shasha**[1]

[1]Courant Institute of Mathematical Sciences, Department of Computer Science, New York University, New York, NY 10012
[2]Center for Genomics and Systems Biology, Department of Biology, New York University, New York, NY 10003
[3]Flatiron Institute, Simons Foundation, New York, NY, 10010, USA
[*]cirrone@courant.nyu.edu

ABSTRACT

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.

Introduction

State-of-the-art methods for gene regulatory network inference[1–4] 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 it 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[5].

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[6]. Unfortunately, genomic interactions are decidedly non-linear, noisy and incomplete[7].

For these reasons, we have approached the causality problem differently: we first attempt to build a model for each gene g that can predict the expression of that gene in left-out time points. If our model is good, then the transcription factors that most influence gene g likely constitute the causal elements 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[8] 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 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 that prior 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[9], that support forecasting and non-linear relationships, but currently lack the ability to incorporate priors. Other time-based machine learning methods such as Inferelator[6] and Dynamic Factor Graph[10], which we used in our previous studies[11, 12] are based on regularized linear regression. We also compare OutPredict with a neural net-based method built to predict gene expression time series[13]. Another relevant time series method from the literature is Granger causality, which has been used successfully for small numbers of genes[14, 15]. Granger causality is a vector autoregressive method that, in that context, could be used to infer important transcription factors. In our case, 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[16], 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.

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[17]. The gold standard network prior are the curated collection of high confidence edges from high throughput ChIP-seq and transcriptomics assays on SubtiWiki[18] (we used the parsed data set provided in[19]).

Arabidopsis thaliana in shoots[12]: 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 validated edges from the plant cell-based *TARGET* assay, which was used to identify direct regulated genome-wide targets of N uptake/assimilation regulators[12].

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)[20]. 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[21]. We retrieved both parsed expression dataset and gold standard data from[9].

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[22]. As gold standard network data, we used the experimentally validated TF-target binding interactions in the DroID database[23]. 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. [9] also used this Drosophila data.

DREAM4 synthetic data[24]: a synthetic dataset from the DREAM4 competition, consisting of 100 genes and 100 TFs (any gene can be a regulator). Because this is synthetic data, the underlying network is known.

Methods

Time series predictions using Random Forests

OutPredict 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, which have been used previously in Genie3[25], iRafNet[26], DynGenie3[9]. They are an ensemble learning approach that can handle linear and non-linear functions and require much less data than neural

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)[17]	52(3reps)[17]	4218	239	3144(154)[19]
Arabidopsis[12]	9(3),9(3)	0	2173	162	1731(7)
E. coli	7(3),7(3),7(3),9(3),5(3)[20]	0	2006	163	4899(163)[9]
Drosophila	28(1)[22]	0	1000	14	1660(9)[23]
DREAM4[24]	20 different time-series with 11 time-points (1rep)	201(1rep)	100	100	176(41)

Table 1. Description of Datasets: the table shows the number of data points in each time series (in parenthesis 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.

networks. 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, . . . , 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 × 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. The parameter settings for the bootstrap is $2/3$, which mean that each tree is built on a bootstrap sample of size $2/3$ of the training dataset. Bootstrap sampling is done with replacement, and the remaining 1/3 for each tree is used to compute the out-of-bag score (validation score).

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 related to pruning of *RandomForestRegressor* in *sklearn*.

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 x 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) $X_1, X_2, \ldots.., X_r$ according to the prior weights (Figure 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[26]. 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).

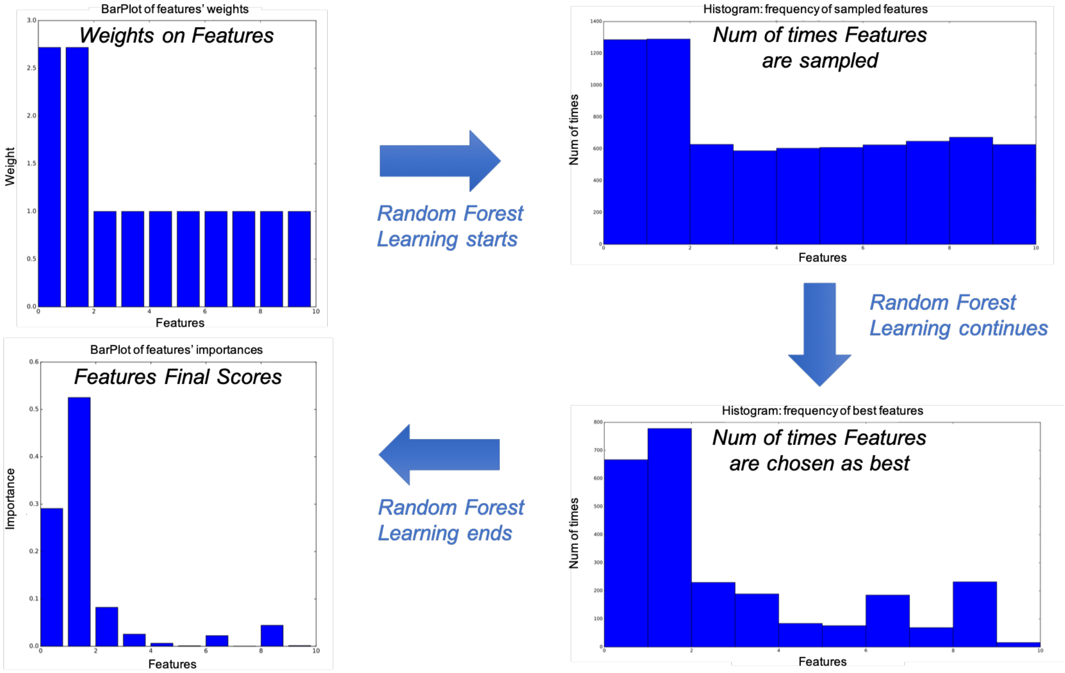

Figure 1. Illustration of how priors work: the priors assign initial weights to features (transcription factors) which influence how likely they are to be chosen as splitting elements in the trees of the Random Forest. As learning takes place, these weights can change, finally leading to a model that 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[26], 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.

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, for each time point t, we seek a function $y_j(t) = fsteady_j(X(t))$ where $X(t)$ must not include 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, we see that the average absolute difference of the gene expression between t and t + 15 minutes decreases as t grows larger in the Arabidopsis time series. Here is a typical data point: the average absolute difference across all genes in Arabidopsis is 56% higher for the interval 5..20 minutes compared to 45..60 minutes. Further, Supplementary Figure S5 illustrates the absolute difference in gene expression decreasing over time for most of the species.

During training, one of the Time-Step or ODE-log models is selected based on the out-of-bag score on the training data. (the out-of-bag score is the same as the Cross-Validation(CV) score for Random Forests). 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 (Supplementary Table S1 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 hyper-parameters. We show in Supplementary Table S2 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[8], 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 experiments in each associated subset. 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.

Algorithm 1 OutPredict Method

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

For the purpose of inferring relative influence of transcription factors on genes and constructing a network of such influence, 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[8]. These importance scores identify the set of the likely most influential transcription factors for each target gene.

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[13] which is an approach developed specifically for time series gene expression prediction (in the supplement). 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[9], which is an extension of Genie3[25] 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[6]. iRafNet[26], 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 cross-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 (Fig 2A, 3A) and calculate the Mean Squared Error (MSE) across all genes.

Quantitative Results

We show in Figure 2B and Figure 3B overall bar plots for a *Bacillus subtilis* and *Arabidopsis*. Similar results hold for other species (Supplementary Figures S1, S2, S3). A table showing which method and data were used for each can be found in Table 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 2A and 3A 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. 2), OutPredict performs 30% better than Penultimate Value ($P < 0.05$), and 50% better than Dynamic Genie3 ($P < 0.05$) (Fig. 2B). As OutPredict 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 (Supplementary Figure S4). In these tests, the inclusion of validated gold-standard edges as priors improved predictions compared to excluding priors (Supplementary Figure S4, 11% improvement, $P < 0.05$).

We show in Table 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 cross-validation (CV) 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. 2B), though less so in Arabidopsis Shoots (Fig. 3B). There is no improvement in E. coli, Drosophila or Dream4 (Supplementary Figs S1, S2, S3). The precise reasons may vary: gold standard data may contain inaccurate regulatory interactions, may be either incomplete, or may depend on specific experimental conditions.

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, totaling 352,026

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[9]
NN	Neural Network	one hidden layer as described in[13]
Pen. Value	Penultimate Value	the second to last time points of each time series is used as the prediction for the last one.

Table 2. Legend of Experimental Results.

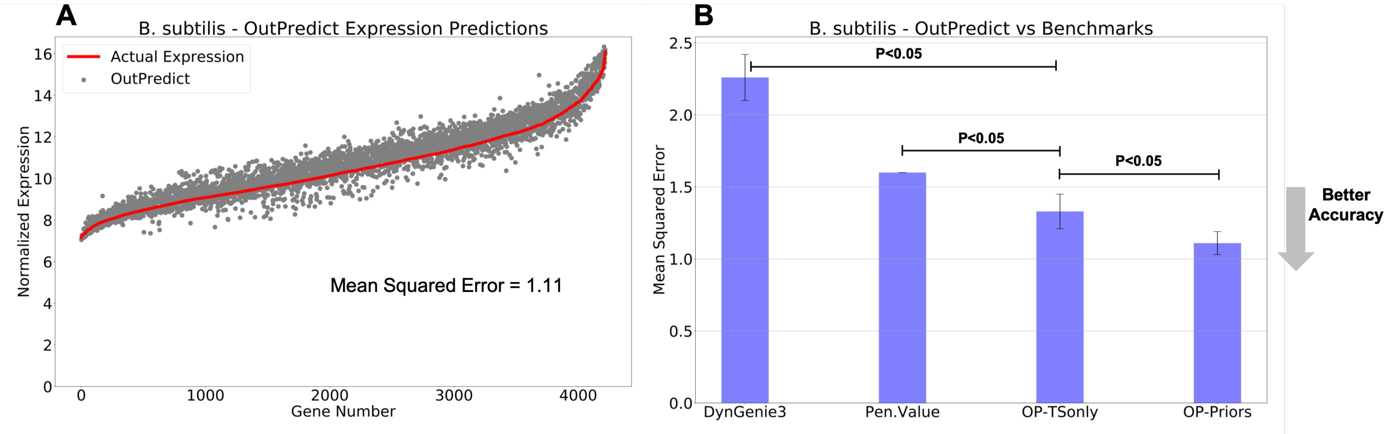

Figure 2. Bacillus subtilis. (A) Comparison of predicted gene expression using OutPredict (grey dots) versus actual expression (red line) at the left-out time point. Genes are ordered by increasing actual mean expression value (red line). OutPredict predicts gene expression well at all expression levels. 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 2. OutPredict (*OP-Priors*) performs significantly better (P<0.05) 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 (Supplementary Table S3); 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) improvement in predictions relative to OutPredict using only time series data. Specifically, prior gold standard data is significantly helpful, showing a 11% relative improvement (Supplementary Figure S4). Finally, Cross-validation (CV) analysis concludes that the Time-step differencing model is better than the ODE-log.

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[27,28,29,30,31,32,33,34] (the data for the 11 TFs are described in Supplementary Table S4). This analysis establishes the precision (i.e., the proportion of 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 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). Moreover, the precision of OP-Priors for the top 2% outperforms OP-TSonly (precision=0.226) and DynGenie3 (precision=0.158). As this proof-of-concept validation shows, OutPredict's importance measures can help to discover potentially causal regulatory relationships. We further compared the performance of the OP-Priors model importances with OP-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

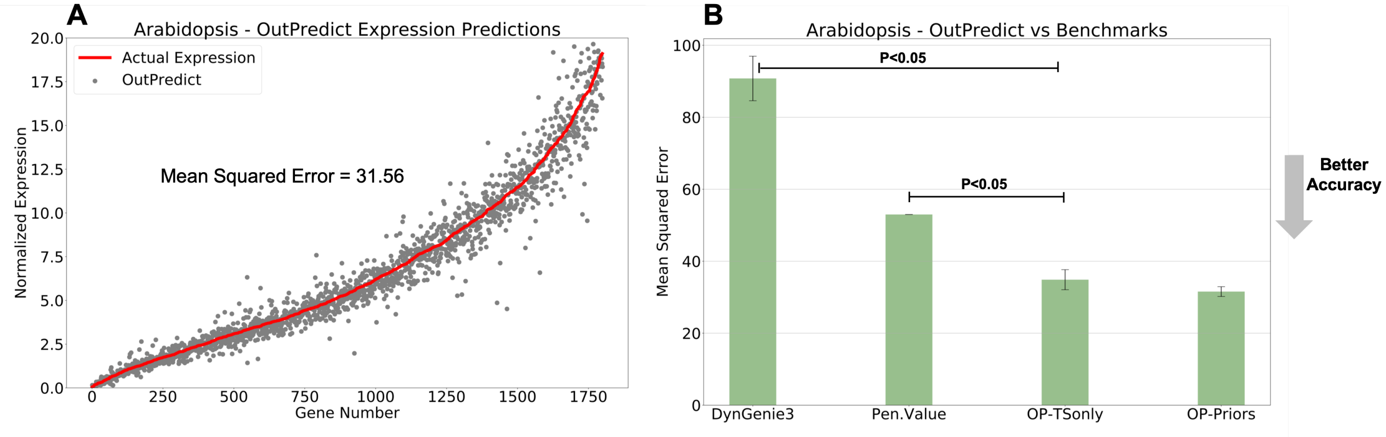

Figure 3. Arabidopsis in Shoot Tissue (time series only dataset) (A) Predicted gene expression using OutPredict (grey dots) compared to actual expression (red line) at the left-out time point. (B) Comparison of time series forecasting: the accuracy of forecasting, measured by Mean Squared Error, has higher values in this case than for other species, because the data is RNAseq and read counts have a broad dynamic range. Table 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), and 61.5% better than Dynamic Genie3 (P < 0.05). 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). The ODE-log model is better than Time-Step based on the cross-validation (i.e., out-of-bag) score. The Neural Network model doesn't converge because the dataset is small.

(OP-Priors) is 15% better than random (p-value < 0.01), for Outpredict without Priors (OP-TSonly) AUPR is 7.5% better than random (p-value < 0.01), while DynGenie3 is no better than random. This shows the promise of using prediction to infer influence (Figure 4).

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. 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.

Discussion

OutPredict 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.

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

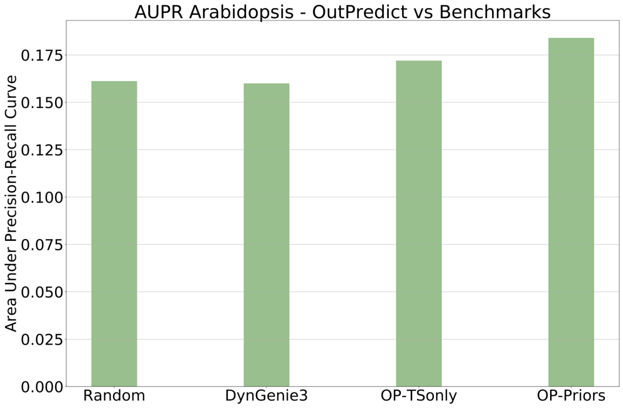

Figure 4. AUPR of Outpredict with Priors (OP-Priors) is 15% better than random (p-value < 0.01); AUPR of Outpredict without Priors (OP-TSonly) is 7.5% better than random (p-value < 0.01); DynGenie3 same as random.

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. The code is open source and is available at the site github.com/jacirrone (*DOI-10.5281/zenodo.3611488*)

Acknowledgements

The authors gratefully acknowledge funding from the following sources: NIH-NIGMS: 1 R01 GM121753-01 to G.M.C and D.E.S, an NIH NIGMS Fellowship 1F32GM116347 to M.D.B, and a grant on Plant Genomics to G.M.C. from the Zegar Family Foundation (A16-0051).

Contributions

J.C., M.D.B., R.B., G.M.C., and D.E.S. designed research, conceived the experiments and reviewed the manuscript. J.C. and M.D.B analyzed the data. J.C. contributed new analytical tools and performed the experiments.

Competing Interests

The authors declare no competing interests.

References

1. Marbach, D. *et al.* Wisdom of crowds for robust gene network inference. *Nat. Methods* (2012).
2. Chai, L. E. *et al.* A review on the computational approaches for gene regulatory network construction. *Comput. Biol. Medicine, 48, 55-65* (2014).
3. Novere, N. L. Quantitative and logic modelling of molecular and gene networks. *Nat. Rev. Genet. 16, 146-158* (2015).
4. Delgado, F. M. & GAmez-Vela, F. Computational methods for gene regulatory networks reconstruction and analysis: A review. *Artif. Intell. Medicine, Vol. 95* (2019).
5. Gitter, A. *et al.* Backup in gene regulatory networks explains differences between binding and knockout results. *Mol. Syst. Biol.* (2009).
6. Greenfield, A., Hafemeister, C. & Bonneau, R. Robust data-driven incorporation of prior knowledge into the inference of dynamic regulatory networks. *Bioinformatics* (2013).
7. Slattery, M. *et al.* Absence of a simple code: how transcription factors read the genome. *Trends Biochem. Sci. 39(9): 381-399* (2014).
8. Breiman, L. Classification and regression trees. *Chapman & Hall CRC* (1984).
9. Huynh-Thu, V. A. & Geurts, P. Dyngenie3: dynamical genie3 for the inference of gene networks from time series expression data. *Sci. Reports* (2018).
10. Mirowski, P. & LeCun, Y. Dynamic factor graphs for time series modeling. *Mach. Learn. Knowl. Discov. Databases, Pt Ii, 5782: 128-43* (2009).
11. Brooks, M. D. *et al.* Network walking charts transcriptional pathways for dynamic nitrogen signaling using validated and predicted genome-wide interactions. *Nat. Commun.* (2019).
12. Varala, K. *et al.* Temporal transcriptional logic of dynamic regulatory networks underlying nitrogen signaling and use in plants. *Proc. Natl. Acad. Sci.* (2018).
13. Smith, M. R., Clement, M., Martinez, T. & Snell, Q. Time series gene expression prediction using neural networks with hidden layers. *BIOT* (2010).
14. Christopher, P. & David, W. How to infer gene networks from expression profiles. *Interface Focus.* (2011).
15. Zou, C. & Feng, J. Granger causality vs. dynamic bayesian network inference: a comparative study. *BMC Bioinforma.* (2009).
16. Maziarz, M. A review of the granger-causality fallacy. *The J. Philos. Econ. Reflections on Econ. Soc. Issues. VIII* (2015).
17. Nicolas, P. *et al.* Condition-dependent transcriptome reveals high-level regulatory architecture in bacillus subtilis. *Science* (2012).
18. Michna, R., Commichau, F., Todter, D., Zschiedrich, C. & Stulke, J. Subtiwiki-a database for the model organism bacillus subtilis that links pathway, interaction and expression information. *Nucleic Acids Res. 42:D692 - D698* (2014).
19. Arrieta-Ortiz, M. L. *et al.* An experimentally supported model of the bacillus subtilis global transcriptional regulatory network. *Mol. Syst. Biol.* (2015).
20. Jozefczuk, S. *et al.* Metabolomic and transcriptomic stress response of escherichia coli. *Mol. Syst. Biol.* (2010).
21. Salgado, H. *et al.* Regulondb v8.0: omics data sets, evolutionary conservation, regulatory phrases, cross-validated gold standards and more. *Nucleic Acids Res. 41, D203-D213* (2013).
22. Hooper, S. D. *et al.* Identification of tightly regulated groups of genes during drosophila melanogaster embryogenesis. *Mol. Syst. Biol.* (2007).
23. Murali, T. *et al.* Droid 2011: a comprehensive, integrated resource for protein, transcription factor, rna and gene interactions for drosophila. *Nucleic Acids Res.* (2011).
24. Greenfield, A., Madar, A., Ostrer, H. & Bonneau, R. Dream4: Combining genetic and dynamic information to identify biological networks and dynamical models). *Ed. by Mark Isalan. PLoS ONE 5 (10). Public Libr. Sci. (PLoS): e13397* (2010).
25. Huynh-Thu, V. A., Irrthum, A., Wehenkel, L. & Geurts, P. Inferring regulatory networks from expression data using tree-based methods. *Ed. by Mark Isalan. PLoS ONE 5 (9). Public Libr. Sci. (PLoS): e12776* (2010).

26. Petralia, F., Wang, P., Yang, J., & Tu, Z. Integrative random forest for gene regulatory network inference). *Bioinforma. 31 (12). Oxf. Univ. Press. (OUP)* (2015).

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

28. Bastakis, E., Hedtke, B., Klermund, C., Grimm, B. & Schwechheimer, C. Llm-domain b-gata transcription factors play multifaceted roles in controlling greening in arabidopsis. *Plant Cell* (2018).

29. Behringer, C., Bastakis, E., Ranftl, Q., Mayer, K. & Schwechheimer, C. Functional diversification within the family of b-gata transcription factors through the leucine-leucine-methionine domain. *Plant Physiol.* (2014).

30. Luo, X. *et al.* Integration of light-and-brassinosteroid signaling pathways by a gata transcription factor in arabidopsis. *Dev. Cell* (2010).

31. Fan, M. *et al.* The bhlh transcription factor hbi1 mediates the trade-off between growth and pathogen-associated molecular pattern-triggered immunity in arabidopsis. *Plant Cell* (2014).

32. Marchive, C. *et al.* Nuclear retention of the transcription factor nlp7 orchestrates the early response to nitrate in plants. *Nat. Commun.* (2013).

33. Gregis, V. *et al.* Identification of pathways directly regulated by short vegetative phase during vegetative and reproductive development in arabidopsis. *Genome Biol.* (2013).

34. Bustos, R. *et al.* A central regulatory system largely controls transcriptional activation and repression responses to phosphate starvation in arabidopsis. *Plos Genet.* (2010).