**Response to Reviewer 1:**

**Major points:**

**Reviewer 1.1.1: How to split the train and test set for what fold (5 fold or 10 fold?) cross validation? If the train and test is for the same gene pair, like trained by T-1 points and tested on the last point, or trained by the whole time points of one gene pair and tested on the whole time points of another gene pair? If it is the former, can the authors explain what is biological meaning of it?**

**Response:** The training and testing sets are strictly separated, and the test set includes only the last time points of each time-series. 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.

Further, 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, that corresponds to a multivariable rather than gene pair model.

To address the reviewer’s questions, we have included this explanation in the subsection “Time series predictions using Random Forest” of the revised manuscript.

**Reviewer 1.1.2:** **What is parameter settings for the bootstrap, like how many times, what is the ratio, are the features also sampled?**

**Response:** 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).

To address the reviewer’s questions, we have extended the description of the algorithm in the subsection “Time series predictions using Random Forest” of the revised manuscript.

**Reviewer 1.1.3:** **What prior knowledge is used for each task, cell-type specific Chip-Seq result or any other dataset?**

**Response:** We have added additional details about the source of the gold standard network used as priors in this study to the Data section of the manuscript. This includes details about the technique(s) used to identify those edges. As the validated edges for E. coli from RegulonDB are a curated collection of interactions from many different sources, we did not list each edge type but refer readers to RegulonDB where they can download the labeled interaction data. For all other:

**B. subtilis**: this dataset consists of time series and steady-state data capturing the response of B. subtilis to a variety of stimuli \cite{Nicolas:2012}. The gold standard network prior is the curated collection of high confidence edges from high throughput ChIP-seq and transcriptomics assays on SubtiWiki\cite{Michna:2014}, we used the parsed data set provided in \cite{Ortiz:2015}.

**Arabidopsis thaliana in shoots** \cite{Varala:2018}:

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\cite{Varala:2018}.

**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)\cite{Jozefczuk:2010}.

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\cite{Salgado:2013}.

We retrieved both parsed expression dataset and gold standard data from \cite{Huynh:2018}.

**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\cite{Hooper:2007}. As gold standard network data, we used the experimentally validated TF-target binding interactions in the DroID database\cite{Murali:2011}. 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.

\cite{Huynh:2018} also used this Drosophila data.

**DREAM4 synthetic data** \cite{Greenfield:2010}: 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.

**Reviewer 1.1.4:** **What is the detailed way to convert the prior knowledge to prior weight?**

**Response:** 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.

When a gold-standard (GS) network (e.g. from ChIP experiments) is used as the prior and there is an edge from a transcription factor TF to g in this network (i.e. true positive edge), then TF is given a high prior weight in the model for gene g. Computationally, we set these prior weights values according to their out-of-bag scores on the training set. The final importance of a TF may increase or decrease relative to its prior value in the course of the training (Figure 1).

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).

To address the reviewer’s question, the revised manuscript now includes a detailed explanation in the subsection “Incorporation of gold-standard data as prior”.

**Reviewer 1.1.5:** **How is the model trained? How to generate the tree, split the node, any pruning to avoid overfitting?**

**Response:** 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. Additionally, the number of features sampled at each node is set to the square root of the total number of transcription factors (i.e., input candidate regulators).

We have clarified this point in the revised text.

**Reviewer 1.1.6:** **Fig. 1 is not good. Why is there only x in the tree model? What does k mean here? What does the black pathway in the tree mean?**

**Response:** We have now removed the former Figure 1 since it was not clear for both reviewers, yet we have extensively described the algorithm in the Methods Section, thus the revised manuscript now includes pseudo-code and an exhaustive explanation.

**Reviewer 1.1.7:** **In the mathematical formulation part, the notation is redundant and unclear, which need to be rewritten.**

**Response:** To address the reviewer’s concern, we have rewritten the mathematical formulation part for clarity and precision.

**Reviewer 1.1.8:** **For the ODE model, what does the ‘log’ mean? Reviewer 1.1.9:** **The authors say 30 mins is similar to 40 mins while different to 2 mins. Any reference for such statement, like kinetics or dynamics parameters measured in other studies?**

**Response to 1.1.8 & 1.1.9:** To address both of reviewer’s points, we have further clarified them in the revised text as follows: 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 changes in inputs tend to have their greatest impact early after the change occurs, so a time interval of say 40 minutes should not give such a different result from a time interval of say 30 minutes, whereas 2 minutes and 12 minutes may give quite different results. We give empirical examples in the text and the supplement.

**Reviewer 1.1.10:** **The equation of I (d) is confusing. What is it used for, splitting tree node in training or ranking important TFs after training?**

**Response:** At a given node d in a tree, OutPredict computes I(d), which is 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. The prior weight from a given feature X\_i to a given target gene y causes features with high prior weights, to be chosen with higher probability, when splitting a tree node during tree construction.

OutPredict calculates the I(d)(variance reduction \* prior weight) criterion (which is defined 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). The selected subset of transcription factors are randomly sampled at each tree node, biased based on the weights of the priors.

We have now clarified this in the revised text to address the reviewer’s question.

**Reviewer 1.1.11:** **Pseudocode for the method is necessary.**

**Response:** The revised manuscript now includes “Algorithm 1 OutPredict Method” related to pseudo-code for the OutPredict procedure.

**Reviewer 1.1.12:** **For comparison, can the authors compare with granger causality methods additionally?**

**Response:** Granger causality is a relevant time series method from the literature. It has been used successfully for small numbers of genes\cite{Penfold:2011, Zou:2009}. 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 infer predictive power using a large number of candidate transcription factors using very short (e.g. 6 time points). As is well known\cite{Mariusz:2015}, 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.

We thank the reviewer for pointing us to this relevant manuscript and have now cited it in our revised manuscript.

**Reviewer 1.1.13: why neural network’s performance is so bad, can the authors try deeper ones?**

**Response:** We don't believe deep neural networks will be good because of the insufficient data, i.e. the number of data points is small relatively to the number of features, and we have confined ourselves to the state-of-the-art method.

Small data sizes relative to the number of causal elements (please refer to columns Number of Points and TFs in Table 1) preclude the use of neural networks and deeper neural network would increase the number of model’s parameters. We have explained this in the text in section …

**Reviewer 1.1.14:** **Besides MSE and precision at low call, can the authors show ROC and PRC results for all methods?**

**Response:** To address this reviewer’s question, the revised manuscript includes additional data analyses we performed to compare the different methods.

We have found that 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 have 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 (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).

**Reviewer 1.1.15:** **It seems that dyngenie3 also used random forest, can the authors explain the reasons why Outpredict is much better than it?**

**Response:** 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 \textit{ODE-log} model, which includes a degradation term that is also cross-validated (unlike all other methods).

In the revised manuscript, we have updated the discussion section to clarify this.

**Minor points:**

**Reviewer 1.2.1: The sota method reference is in 2012, can the authors find some latest ones?**

**Response:** In addition to \cite{Marbach:2012}, we have added more recent references to address the reviewer’s question.

**Reviewer 1.2.2:** **It would be better if the reference were put into Table 1.**

**Response:** In the revised manuscript**,** we have added the references to Table 1 as well in order to address the reviewer’s point.

**Reviewer 1.2.3:** **the font size in figures should be consistent.**

**Response:** We have now carefully adjusted the figures to address the reviewer’s point.

**Reviewer 1.2.4:** **What do OP, TS, SS mean in page 6?**

**Response:** We have now further specified that OP, TS, and SS stand for OutPredict, Time Series and Steady-State.

**Reviewer 1.2.5:** **Fig.3A legend has grammar error.**

**Response:** We have not been able to find a grammar error in the legend of what is now Fig. 2A, nevertheless we have slightly changed the wording.

**Response to Reviewer 2:**

**Reviewer 2.1.1:**  **One of the main messages of the paper (as stated in the title, the abstract and the introduction) is that OutPredict, in addition to predicting gene expression, is also able to infer causal edges between genes. I think this statement is too strong, as this was evaluated only on a single dataset, without any comparison to other existing network inference methods. The performance of OP-Priors should be at least compared to OP-TSonly and DynGenie3 (as it was done in the rest of the results section).**

**Response:** To address this reviewer’s question, the revised manuscript includes additional data analyses we performed to compare the different methods.

We have found that 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 have 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 (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).

**Reviewer 2.1.2:** **The OutPredict method is not clearly described, and Figure 1 does not bring much information. How are the prior weights constructed from steady-state data exactly? It is also not clear how the prior weights are taken into account during the construction of the trees. Is a subset of variables randomly selected at each tree node, with a bias towards the priors (as done in iRafNet)? In that case, how many variables are selected at each node? Or is the I(d) (variance reduction \* prior weight) criterion calculated for all the variables? Finally, when multiple prior datasets are available, how are the prior weights combined?**

**Response:** 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.

When a gold-standard (GS) network (e.g. from ChIP experiments) is used as the prior and there is an edge from a transcription factor TF to g in this network (i.e. true positive edge), then TF is given a high prior weight in the model for gene g. Computationally, we set these prior weights values according to their out-of-bag scores on the training set. The final importance of a TF may increase or decrease relative to its prior value in the course of the training (Figure 1).

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 tree construction, OutPredict’s Weighted Random Forest, at each node d, selects r candidate features (i.e., transcription factors) X\_1, X\_2, ....., X\_r according to the prior weights; 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\cite{Petralia:2015}. In addition, OutPredict calculates the I(d) (variance reduction \* prior weight) criterion (which is defined 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).

OutPredict does not combine the prior weights from multiple datasets when there is steady-state and gold standard network data available, but it 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\cite{Petralia:2015}, 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.

We have now removed the former Figure 1 since it was not clear for both reviewers, yet we have extensively described the algorithm in the Methods Section, thus the revised manuscript now includes pseudo-code and an exhaustive explanation.

**Reviewer 2.1.3: The authors should specify the values of the hyper-parameters (alpha, prior weights) that are tested when calculating the out-of-bag score.**

**Response:** To address the reviewer’s question, we show, in Supplementary Table S2 of the revised manuscript, the set of values tested for the degradation term alpha and for the prior weights when calculating the out-of-bag score.

When OP-Priors is set to True, and gold standard data is given as priors, in order to compute prior weights from the gold standard prior knowledge, we assign 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.

**Reviewer 2.1.4: It would be interesting to have more detailed results that compare the performance of the Time-Step model to the ODE-log model. For example, it would be interesting to see, for a given organism, how many times one model is selected over the other. Is one of the model clearly more performant across the different genes, and does it change a lot from one organism to another?**

**Response:** To address the reviewer’s question, we have added a new Supplementary Table (S1) to the revised manuscript. For a given organism the table shows the best model based on out-of-bag score between Time-Step (TS) and ODE-log. 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. We determine this on the training data and then apply whichever method is better on the test data.

**Reviewer 2.1.5: The MSE of OP-Priors is significantly lower than the MSE of OP-TSonly for only one dataset (B. subtilis). The authors should discuss this result. Is is due to the methodology used? Or maybe the quality of the priors?**

**Response:** 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. We have clarified this in the revised manuscript.

**Minor comments:**

**Reviewer 2.2.1:** **The ODE-log model should be a bit more explained (the parameter alpha is not even defined).**

**Response:** The parameter alpha is the degradation term and all the genes are assumed to have the same alpha. Further, to address the reviewer’s point, we have updated the mathematical formulation section to detail and clarify our procedures regarding ODE-log model.

**Reviewer 2.2.2: In the definition of the importance score s\_i (bottom of page 4), I suppose that the sum is over the nodes where the variable X\_i was selected?**

**Response:** Let D\_i be the set of nodes which branch based on transcription factor (feature) X\_i, 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 \cite{breiman:1984}. These importance scores identify the set of the likely most influential transcription factors for each target gene.

In the revised manuscript, we have updated the Mathematical Formulation part to detail this algorithm’s procedure.

**Reviewer 2.2.3:** **How is the MSE calculated when there are several replicates for the last time point? Is the MSE averaged over the different replicates?**

**Response:** 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. We have clarified this in the revised manuscript.

**Reviewer 2.2.4:** **Some details about the hyper-parameters of the neural network model should be given (e.g. number of neurons in the hidden layer, regularization or not, etc.).**

**Response**: The Neural Network (NN) model with a hidden layer\cite{Smith:2010} we used 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. The revised manuscript includes now these further details about the neural network model.