Bioinformatics, 36(8), 2020, 2522–2529
doi: 10.1093/bioinformatics/btz950
Advance Access Publication Date: 6 January 2020
Original Paper

Systems biology

LiPLike: towards gene regulatory network predictions of high certainty

Rasmus Magnusson * and Mika Gustafsson*

Department of Physics, Chemistry and Biology, Linköping University, Linköping 581 83, Sweden

*To whom correspondence should be addressed.

Associate Editor: Inanc Birol

Received on May 29, 2019; revised on December 5, 2019; editorial decision on December 18, 2019; accepted on January 3, 2020

Abstract

Motivation: High correlation in expression between regulatory elements is a persistent obstacle for the reverse-engineering of gene regulatory networks. If two potential regulators have matching expression patterns, it becomes challenging to differentiate between them, thus increasing the risk of false positive identifications.

Results: To allow for gene regulation predictions of high confidence, we propose a novel method, the Linear Profile Likelihood (LiPLike), that assumes a regression model and iteratively searches for interactions that cannot be replaced by a linear combination of other predictors. To compare the performance of LiPLike with other available inference methods, we benchmarked LiPLike using three independent datasets from the Dialogue on Reverse Engineering Assessment and Methods 5 (DREAM5) network inference challenge. We found that LiPLike could be used to stratify predictions of other inference tools, and when applied to the predictions of DREAM5 participants, we observed an average improvement in accuracy of $>$140% compared to individual methods. Furthermore, LiPLike was able to independently predict networks better than all DREAM5 participants when applied to biological data. When predicting the *Escherichia coli* network, LiPLike had an accuracy of 0.38 for the top-ranked 100 interactions, whereas the corresponding DREAM5 consensus model yielded an accuracy of 0.11.

Availability and implementation: We made LiPLike available to the community as a Python toolbox, available at https://gitlab.com/Gustafsson-lab/liplike. We believe that LiPLike will be used for high confidence predictions in studies where individual model interactions are of high importance, and to remove false positive predictions made by other state-of-the-art gene–gene regulation prediction tools.

Contact: rasmus.magnusson@liu.se or mika.gustafsson@liu.se

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Understanding and interpreting big data have become a focal point of the field of bioinformatics, since several scientific areas, such as preclinical medicine, are dependent on a thorough understanding of biological processes. The studies have been fuelled by an increasing stream of data from several omics' techniques. The overwhelming size and complexity of data from modern techniques, such as RNA-Seq hinder the drawing of conclusions from straightforward analyses (Magnusson *et al.*, 2017). Therefore, network analysis has emerged as a prominent tool that is used to both distinguish and interpret cellular processes (Alvarez *et al.*, 2018; Cheng *et al.*, 2018; Madar *et al.*, 2010; Prill *et al.*, 2010; Santolini and Barabási, 2018). In such network analyses, the aim has typically been to infer the molecular functions behind biological processes. These molecular functions have often been in the form of gene expression regulation, and the complex network that forms between gene transcription regulators and dependent genes (Zhang *et al.*, 2015). This network analysis can be performed in several ways, such as by constructing graphs of nodes and edges that give information about cellular processes. When reverse-engineering gene regulatory networks (GRNs), genes are denoted as nodes while the aim is to infer interactions between them, referred to as edges. Next, studying the GRN can provide different insights into the biological system (Madhamshettiwar *et al.*, 2012; Wang *et al.*, 2019). These insights include identifications of important feedback or cross-talks in biological systems (Magnusson *et al.*, 2017), understanding of upstream master regulators of gene expression (Gustafsson *et al.*, 2015; Lefebvre *et al.*, 2010) and potential drug targets (Guney *et al.*, 2016; Madhamshettiwar *et al.*, 2012).

Currently, there are several methods that are commonly used to predict GRNs from gene expression data. One such approach is to assume the data to be generated from a linear system and apply regression-based algorithms. Such methods include the LASSO (Tibshirani, 1991), the Elastic Net (Zou and Hastie, 2005) and the Inferelator (Bonneau *et al.*, 2006; Madar *et al.*, 2009). Other popular methods include mutual information, notably, ARACNe (Margolin *et al.*, 2006), Bayesian modelling (Friedman *et al.*, 2000;

Kwon, 2016), artificial neural networks (Marbach *et al.*, 2012) and methods based on correlation (Xiong and Zhou, 2012).

Although there are a number of cases where GRN inference has been successfully applied to solve biological questions (Noh *et al.*, 2018; Madhamshettiwar *et al.*, 2012; Gustafsson *et al.*, 2015; Guney *et al.*, 2016), network analysis has historically struggled with a set of problems. First, the results of network inference are often dependent on the underlying quality of data, with factors, such as correlated regulators posing a problem for most algorithms (Barzel and Barabási, 2013; Tjärnberg *et al.*, 2015; Zhao and Yu, 2006). Second, understanding what aspects of the results can be trusted often poses a problem, as most network inference methods aim to recreate the most probable network, which often comes at the expense of a high false prediction rate. An example of these low accuracies can be found in the prestigious Dialogue on Reverse Engineering Assessment and Methods 5 (DREAM5) GRN prediction challenge, a contest where participants were given gene expression datasets from four independent sources with the objective of predicting gene regulation. Analysing the results of DREAM5, the areas under precision recall curves were found to be around 10% for biological networks (Marbach *et al.*, 2012). Such low performances pose a problem for interpretations of GRNs and ultimately hinder biological advances. Biological experiments are both time-consuming and costly and performing experiments, such as drug screening based on network predictions with an accuracy of 10% is ineffective.

However, there are some methods that have addressed the problem of low accuracy in predictions of gene regulation. One commonly applied approach to address the problem of correlated explanatory variables is to apply repeated subsampling of data followed by model training (Morgan *et al.*, 2018; Wang *et al.*, 2011) commonly referred to as bootstrapping. However, such approaches will not fully avoid identifying edges that cannot be rejected and instead will include identifications across correlated explanatory variables (Wang *et al.*, 2011). In two simultaneously developed works by Feizi *et al.* (2013) and Barzel and Barabási (2013), it was shown that predictions stemming from indirect correlations between possible interactions can be removed from an inferred network. Approaches of this type are of special importance in the reverse-engineering of GRNs. Consider a set of three genes, A, B and C, where the expression of B and C are regulated by A. In their gene expression values, B and C will correlate, and it is not simple to based on data distinguish if there is a link from A to C or from B to C. Moreover, it is not certain whether A holds more explanatory power over C than what B does, and in an approach to only identify interactions of high importance, it might be better not to identify any of the interactions in this example if they are not uniquely inferable from data. One such approach is the Robust Network Inference, RNI tool (Nordling, 2013), which has been developed to omit identifications that cannot be rejected by any model (Tjärnberg *et al.*, 2015). However, RNI is not, as of the date of publication, a publicly available tool (Tjärnberg *et al.*, 2015). Furthermore, RNI has been shown to be much stringent, with few predictions being made when the signal-to-noise ratio is low (Morgan *et al.*, 2018).

In this study, we present the LInear Profile LIKElihood (LiPLike) method, a novel algorithm to predict gene to gene regulation with high accuracy by only selecting interactions that are uniquely inferred by measured data. LiPLike was conceptualized by merging the profile likelihood method for parameter confidence estimation (Kreutz *et al.*, 2013) and GRN inference but with two major differences. First, LiPLike assumes a linear regression model, which allows for substantially larger networks to be analysed. Second, LiPLike does not search for a confidence interval for a certain parameter, but instead only compares two key points of interest: the parameter value that minimizes the residual sum of squares, and the point where this parameter equals zero. If there is an alternative linear combination of the explanatory variables that can model a dependent variable equally well, LiPLike will refrain from including such a parameter. In other words, LiPLike will only infer gene regulations with high confidence, and will avoid identifying edges that are not well-determined from data. In the case of several highly correlated explanatory variables that all can individually explain the expression data of a target, LiPLike makes no edge prediction, whereas the LASSO would typically include one at random (Xu *et al.*, 2012) and the Elastic Net would include all (Zou and Hastie, 2005).

We tested the sensitivity of LiPLike by applying it to data from the DREAM5 challenge and comparing the predicted network with the 36 GRN inference methods analysed in the challenge. Instead of assessing the performance of a method based on the precision recall- or receiver operating characteristic curves, we evaluated the accuracy of top-ranked predicted edges for both LiPLike and the DREAM5 participants. We found LiPLike to have higher accuracy than all DREAM5 methods in the biological networks, and better-than-average accuracy when benchmarking against the *in silico* generated data. Moreover, we found that LiPLike could successfully remove false positive identifications from GRN predictions of other methods, and recognized this feature to be useful whenever high accuracy GRN predictions are sought from gene expression data. Finally, to make LiPLike available to the community we built a Python toolbox available for download at https://gitlab.com/Gustafsson-lab/liplike.

2 Materials and methods

2.1 Problem description

LiPLike is a novel GRN inference method that maximizes the accuracy of gene regulatory predictions. LiPLike minimizes the false positive prediction rate by not identifying potential regulators where a regulation can be replaced by a linear combination of one or several other explanatory variables (Fig. 1A). Specifically, the LiPLike algorithm was inspired by the profile likelihood method, used to estimate confidence intervals of estimated model parameters. Consider a system of independent variables X and dependent variables Y.

$$Y = f(X, \beta);\ Y, X, \beta \in \mathbb{R} \tag{1}$$

In Equation (1), $f(X, \beta)$ is a function of parameters mapping X to Y, via parameter vector β. The vector β typically determines how each independent variable quantitatively models Y. In biology, this relationship could, for instance, be transcription factors (TFs) X regulating a target gene Y, with the effects of parameter vector β. The profile likelihood method estimates a confidence interval of a parameter β_i by observing the residuals between Y and $f(X, \beta)$ when β_i is forced to take the user-defined value ζ, as shown in Equation (2).

$$-LL(Y - f(X, \beta|_{\beta_i=\zeta})) < T \tag{2}$$

In Equation (2), the log-likelihood function, LL, is studied with respect to one parameter β_i. By varying ζ, a confidence interval can be defined as the values of ζ for which the negative log-likelihood is below a threshold T. In other words, the profile likelihood tests which value a parameter β_i can take while still being able to explain a variable Y. For a detailed description of the profile likelihood approach, see Kreutz *et al.* (2013). The profile likelihood is, however, mostly applied to non-linear systems and for grey-box models where all parameters are known to exist. Here, we instead assume that data are generated from a linear system of variables Y dependent on variables X via a vector of unknown constants, β.

$$Y_{j,k} = \sum_i \beta_{i,j} X_{i,k} + \varepsilon \tag{3}$$

In the annotations of Equation (3), $Y_{j,k}$ is a scalar corresponding to the expression of gene j at observation k. Likewise, $X_{i,k}$ is a scalar corresponding to regulator i at observation k. As is widely known, for overdetermined systems the vector β can be analytically estimated, via the methods of least squares, to minimize the sum of squared residuals. Now, for each parameter $\beta_{i,j}$, there are two parameter values that are of interest, namely the one minimizing the residual sum of squares and the point where $\beta_{i,j} = 0$. Thus, we can quantify the relationship between these two points and introduce the term $q_{i,j}$.

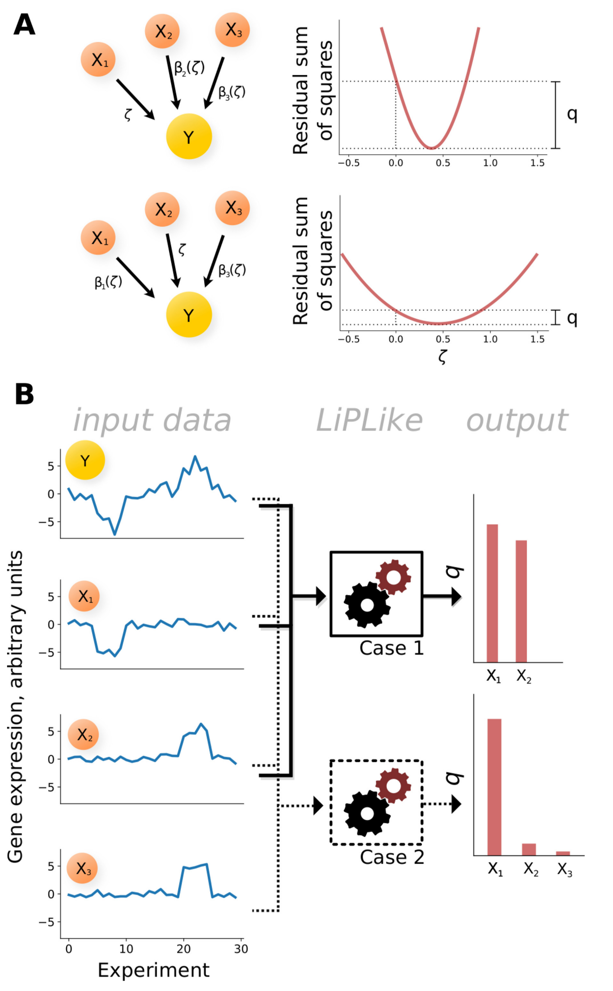

Fig. 1. Illustration of LiPLike rationale. (**A**) In a toy system of three gene regulators (X_1, and the correlated variables X_2 and X_3) regulating a target gene (Y), the optimal parameters of the corresponding linear model are easily identified using the method of least squares. Next, by imposing a constraint that the parameter value of X_1 should equal a value ζ and iteratively re-estimating the remaining two variables, the profile of the residual sum of squares as a function of ζ can be studied. In other words, as the parameter ζ is changed, so will the ability of the regulators to explain the data in Y. Of special interest is the point of $\zeta = 0$, i.e. where the regulator is removed from the system. Furthermore, if X_1 is uniquely needed to model Y, the residual sum of squares as a function of ζ will increase rapidly, as seen in the top case, and there will be a large increase in the residual sum of squares between the best fit and the case where $\zeta = 0$. In the bottom case, since X_2 and X_3 are correlated, there exists a linear combination of remaining explanatory variables that can adequately fit Y, and the residual sum of squares is less dependent on ζ. This is because when the parameter between X_2 and Y is changed, the variable X_3 is able to take the place of X_2. (**B**) Two examples of LiPLike applied to data. Three independent variables exist, whereof two (X_2 and X_3) have a high correlation between them. To explain dependent variable Y, either (X_1, X_2) or (X_1, X_3) are needed, and there is no way to infer whether X_2 or X_3 is the correct regulator. If X_2 or X_3 is left out from the set, LiPLike infers both remaining inputs to be important, as illustrated by the magnitudes of q shown to the right. When all three independent variables are included, LiPLike refrains from selecting variables that cannot be inferred uniquely. This is because there is no way to determine if X_2 or X_3 is the correct regulator

$$q_{i,j} = \frac{\min_{(\beta|\beta_{i,j}=0)} \sum_k \left(Y_{j,k} - \sum_m \beta_{m,j} X_{m,k}\right)^2}{\min_\beta \sum_k \left(Y_{j,k} - \sum_m \beta_{m,j} X_{m,k}\right)^2} \qquad (4)$$

Given the values of X and Y, LiPLike returns $q_{i,j}$ for all interactions in the network (Algorithm 1). As written in Equation (4), $q_{i,j}$ will take larger values for edges that are uniquely needed to explain Y_j (Fig. 1B). In practice, Equation (4) compares the residual sum of squares of two ordinary least-squares regression problems; one with all dependent variables and one with dependent variable j removed.

Algorithm 1. LiPLike pseudo-code

```
Data: matrices of dependent variables Y and independent variables
        X
Result: Scalar $q$ indicating uniquely defined interaction
for every dependent variable $j$ do
    for every independent variable $i$ do
        $q_{i,j}$ = ($j$:th residual sum of squares of full network except
         independent variable $i$)/($j$:th residual sum of squares of full model);
    end
end
```

If there exists a solution to adequately model response variable Y_j without dependent variable j, the ratio of residual sum of squares $q_{i,j}$ will be close to one. Alternatively, if the removed variable was uniquely needed to fit data in the regression model, the relative increase in the residual sum of squares will be larger. This equation can also be explained with the help of an example. In the case of two highly correlated regulators, X_A and X_B, the constraint $\beta_{a,j} = 0$ will not independently give a considerable increase in the residual sum of squares. Thus, $q_{a,j} \approx 1$, as $\beta_{b,j}$ will adjust accordingly. In Equation (4), there are three properties that should be noted. First, if the system is not overdetermined, i.e. if rank(X) $\leq N$, the denominator of Equation (4) equals zero, and thus q is undefined $\forall i, j$. To solve this problem, we implemented functionality such that LiPLike either takes a prior interaction matrix as the input or builds one based on a cut-off on the Pearson correlation between variables. It should be noted, however, that this functionality interferes with the aim of LiPLike, i.e. to only identify edges that cannot be replaced by *any* other interaction, and we thus recommend to use prior knowledge sparingly. Second, when computing $q_{i,j}$, the residual sum of squares of the special case of $\beta_{i,j} = 0$ is normed to the least-squares fit of the fully connected system. Thus, in the case of a poor general fit to a dependent variable the increase in the numerator must thus be larger than for a well-fitted dependent variable for an edge to be identified. Third, the output of LiPLike $q_{i,j}$, depends only on data properties and there are no additional hyperparameters that affect the performance of LiPLike.

2.2 Package design

We implemented the LiPLike algorithm as a Python toolbox based solely on built-in Python3 functions and the widely used NumPy package. LiPLike calculates ratios of the residual sum of squares, denoted q. The q matrix will hold continuous values $q_{i,j} > 1$, with larger elements indicating an interaction that uniquely explains data. To choose a cut-off for q, i.e. to choose which edges should be considered to be identified, we implemented an optional Monte Carlo simulation of LiPLike applied on data random data. By default, LiPLike draws data from X and Y a 100 000 times and searches for the largest value of q that can be expected to be generated given random chance. Next, we only considered LiPLike to have identified an edge where $q_{i,j}$ takes a value outside the range of the random q values.

3 Results

3.1 LiPLike predicts edges with high accuracy

To test the ability of LiPLike to extract gene–gene interaction predictions, it was first applied to two versions of an *in silico* generated gene expression dataset from the tool geneSPIDER (Tjärnberg *et al.*, 2017). The network contained 100 nodes genes with 300

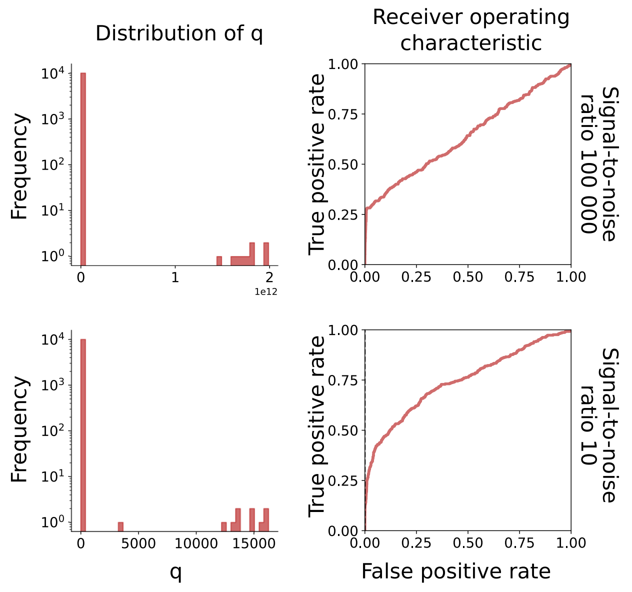

Fig. 2. LiPLike properties and performance on *in silico* generated networks. The confidence of inferred edges is listed as q, calculated for two datasets for the same network. The networks differed in the signal-to-noise ratio. The magnitudes of q were found to be dependent on the noise level, with a factor 10e^7 differing between the datasets, as seen on the x-axis on the histograms to the left. Moreover, the histograms both display a property empirically arising from LiPLike for networks with strong signals, i.e. a separation of confidence into two distinct groups, high confidence or none. To the right are the corresponding receiver operating characteristic curves, showing that LiPLike infers edges well for some values, to then have a near to random chance for identifying an edge. The networks were retrieved from https://bitbucket.org/sonnhammergrni/genespider

measurements at signal-to-noise ratio of 10 and 100 000, respectively. We observed LiPLike to split edges into two groups, a group containing the majority of genes where no high precision prediction could be made ($q_{i,j} \approx 1$), and a minor group where we had a clear indication of an interaction ($q_{i,j} \gg 1$) (Fig. 2, left).

As expected, we found that the spread between these two groups of edges appeared to be dependent on the signal-to-noise ratio of the data, with an increase of q_{max} in order of 10^8 when going from a signal-to-noise ratio of 10 to 100 000. Next, we studied the receiver operating characteristic curves for the respective datasets and observed a behaviour where the true precision rate was initially high, corresponding to a high accuracy. Next, the receiver operating characteristic curves transitioned into a linear phase, i.e. a phase where true predictions were randomly distributed. This linearity indicated that, as expected from Equation (4), LiPLike struggles to predict edges outside the high accuracy interval (Fig. 2, right).

We further aimed at analysing the ability of LiPLike to handle under-determined systems, and sampled 50 experiments 100 times from the data a signal-to-noise ratio (SNR) = 10, and applied LiPLike. By default, when applied to an under-determined network, LiPLike creates a prior network based on Pearson correlation between regulators and targets. In particular, for each gene LiPLike allows for n potential regulators, where n is 90% of the number of available observations. Using this approach, we observed a mean area under the receiver operating characteristic curve of 0.58, as compared to 0.75 when using the full dataset of 300 observations. This example showed how an over-determined system is not necessary for LiPLike to produce usable results, and that LiPLike can still be used when data are scarce.

3.2 LiPLike outperforms state-of-the-art methods in terms of accuracy

Knowing that LiPLike was capable of producing accurate predictions, we next aimed to benchmark LiPLike to other state-of-the-art GRN inference methods. There are several available datasets for benchmarking, of which the DREAM5 network inference challenge is one of the most extensive. Moreover, the DREAM5 data contain ranked predictions of networks from both challenge participants, state-of-the-art methods and a combined crowd estimate, all based on data from four different networks. The networks are based on two prokaryotic (*Staphylococcus aureus* and *Escherichia coli*), one eukaryotic (*Saccharomyces cerevisiae*) and one *in silico* simulated system, all with accompanying expression datasets. However, due to the inability of DREAM5 participants to infer edges from the *S.cerevisiae* data (Marbach *et al.*, 2012), we excluded it from further analysis. We assumed the expression of target genes to be a linear combination of the TFs in the data. Since the rationale of LiPLike is not to predict full networks, we measured performance in terms of accuracy, i.e. the rate of correct predictions among these top-ranked edges and the corresponding number of top-ranked edges in the DREAM5 participants' networks. We identified the *E.coli* network to be of special importance since the *S.aureus* gold standard was less extensive (only 518 edges), and since *in silico* is less relevant than true biologically derived networks.

Studying the performance of LiPLike on the *E.coli* data, we found accuracies of 0.38, 0.27 and 0.18 for 100, 500 and 1000 included edges, respectively (Fig. 3A and B). These numbers can be compared to the consensus model of all DREAM5 participants (Marbach *et al.*, 2012), with corresponding accuracies of 0.11, 0.10 and 0.08, i.e. less than half of LiPLike. Indeed, we found LiPLike to give the highest accuracy of all methods on the interval of 41–7943 of top-ranked edges. To choose a threshold of included edges, we performed a Monte Carlo re-sampling of data (Section 2) and found 2308 gene regulations to have $q_{i,j}$ larger than the max of the random q (*S.aureus*: 263 and *in silico*: 2203). Studying the predictions of both DREAM5 biological networks, we found LiPLike to have a higher accuracy than all DREAM5 participants at the Monte Carlo threshold (LiPLike accuracy = 0.11, 0.10, for *S.aureus* and *E.coli*, which is an increase of 11% and 18%, respectively compared to the best method in the DREAM5 challenge) (Fig. 3C). Notably, applied to the biological datasets, we found LiPLike to outperform the accuracy of the community predictions, which previously have been shown to be successful predictors of GRNs (Marbach *et al.*, 2012). In the *in silico* generated data, we observed LiPLike to only achieve a higher accuracy than the average of the DREAM5 participants. It should be noted, however, that most methods performed well on this data, and the edges predicted by LiPLike still had an accuracy of 0.37 (average = 0.32). We further analysed the recall, i.e. the percentage of the edges in the gold standard that were correctly identified in the LiPLike top edges, and found the recalls to be 6%, 11% and 32% for the *S.aureus*, *E.coli* and *in silico* networks, respectively. With these results, we concluded that LiPLike, used as a standalone tool, is an effective method to extract gene–gene interactions with high confidence.

3.3 LiPLike removed false positives from predictions of other methods

A major conclusion drawn from the DREAM5 challenge was that the union of method predictions robustly outperforms individual methods. We, therefore, examined the impact of combining LiPLike with DREAM5 participants' predictions. For the edges predicted by LiPLike and the same number of top-ranked edges in the community prediction (263, 2308, 2203 edges), we found significant overlaps (odds ratio = 223, 288, 192, Fisher's exact test $P < 10^{-74}$, 10^{-100}, 10^{-100} for the *S.aureus*, *E.coli* and *in silico* networks, respectively). Indeed, we observed significant overlaps of edge predictions between LiPLike and almost all DREAM5 participants (Supplementary Fig. S1), meaning that LiPLike identifies a subsection of interactions that are shared across several network inference algorithms.

Since the rationale of LiPLike is not to identify any edge where there is more than one potential regulator, we further tested to combine LiPLike with other methods for better accuracy and studied the DREAM5 community and LiPLike top-ranked predictions in depth. For the predictions in common for both methods, we found the

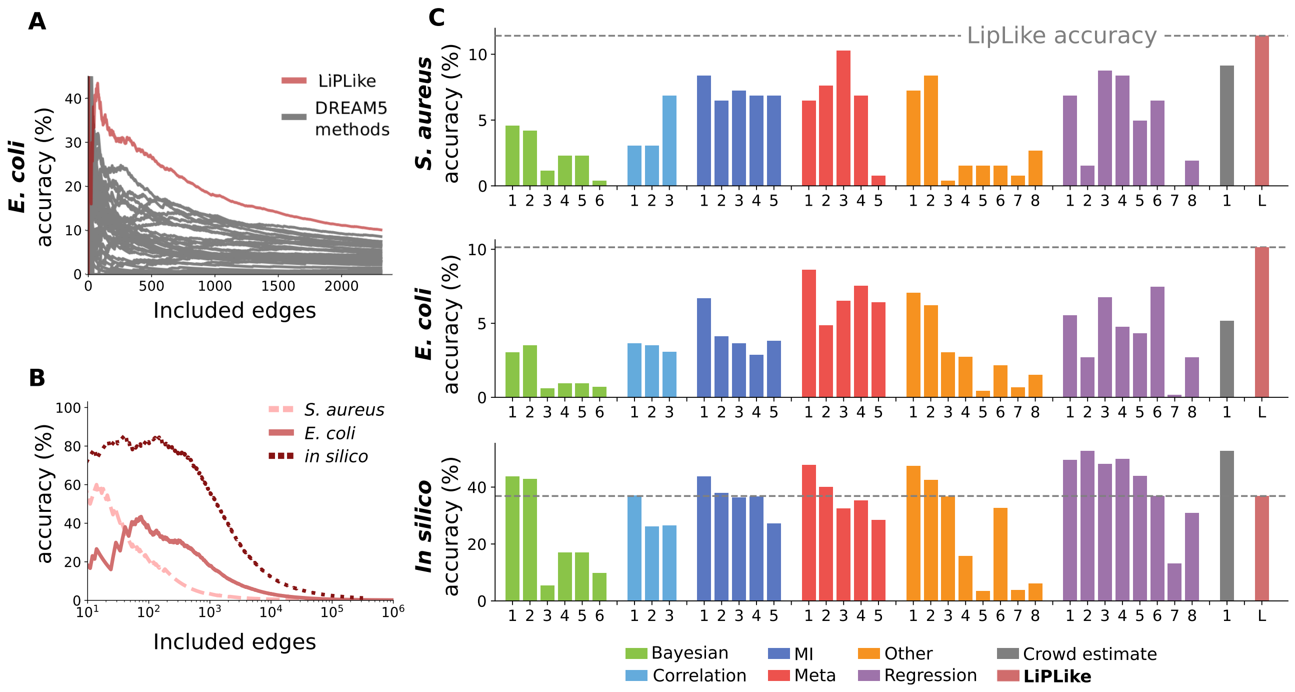

Fig. 3. LiPLike performance on DREAM5 challenge data. (**A**) Accuracy of algorithms predicting edges of the *E.coli* network as a function of number of edges considered. The LiPLike performance is plotted in red, showing a higher accuracy than all DREAM5 participants. (**B**) The accuracies of LiPLike across top-ranked edges for all networks. (**C**) The accuracy of all methods, the crowd estimate and LiPLike for the top-ranked edges. LiPLike gave the highest accuracy of all methods in both biologically derived networks, and ranked 20th of 36 in the *in silico* network. (Color version of this figure is available at *Bioinformatics* online.)

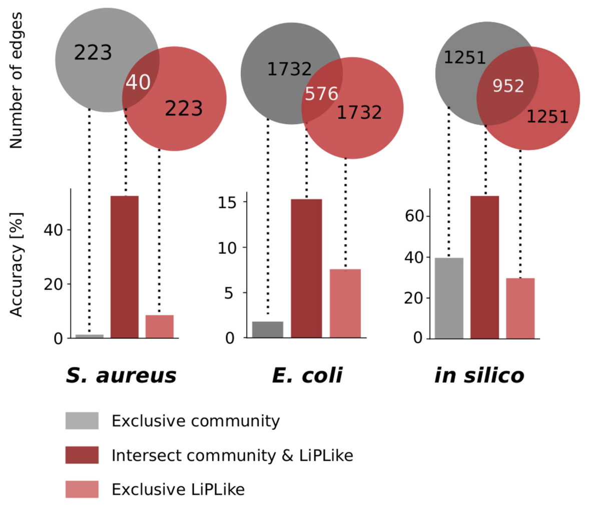

Fig. 4. Accuracy of edge predictions of the DREAM5 community prediction and LiPLike, split up between top edges that are exclusively found in the community, LiPLike, and in both. In all cases, the edges that are found in both predictions have a considerable increase in accuracy compared to the DREAM5 challenge community prediction. Moreover, in the case of the biological networks, *S.aureus* and *E.coli*, LiPLike performs better than the community in the non-overlapping predictions, indicating that LiPLike identifies edges that the community failed to include

community accuracies to increase from 9%, 5% and 53% to 53%, 15% and 70% (Fisher's exact test $P < 1.8 \cdot 10^{-14}$, $5.4 \cdot 10^{-18}$ $2.6 \cdot 10^{-78}$ for *S.aureus*, *E.coli* and *in silico*, respectively; Fig. 4). Moreover, for the non-overlapping edges between LiPLike and the community approach in the *E.coli* and *S.aureus* networks, we also observed an almost complete depletion of correct edges predicted by the community but not by LiPLike, with accuracies as low as 1.8% and 1.3%, respectively (Fig. 4), suggesting that combining LiPLike with other methods can effectively remove false positive edge identifications from a set of predictions. Furthermore, this property of removing false positive interactions seems to be a general property of LiPLike. By examining the intersections between all DREAM5 participants and LiPLike, we found the accuracy to be increased for almost all methods, with median increases in accuracy from 0.05, 0.04 and 0.36 to 0.47, 0.15 and 0.68, for *S.aureus*, *E.coli* and *in silico* (Supplementary Fig. S2). Furthermore, we verified this behaviour by applying LiPLike to independent *in silico* generated data (Prill *et al.*, 2010, Supplementary Material S3). When applied to the expression sets of 10, 50 and 100 genes, we observed LiPLike to remove false positive predictions of two independent network inference methods presented in Zhang *et al.* (2015) and Aghdam *et al.* (2015). This increase in accuracy demonstrated that LiPLike can be used to remove false interaction predictions from other methods. Thus, there is a strong indication that LiPLike can be used in combination of any GRN inference method to stratify predictions into two groups of more and less reliable interactions. We also noted LiPLike to produce accuracies and F_1 measures in line with the other methods (accuracies = 0.43, 0.16, 0.11, F_1=0.35, 0.19, 0.12), when applied as a stand-alone tool to the 10, 50 and 100 gene networks, much like the results from the study of the DREAM5 *in silico* data.

3.4 LiPLike robustly identifies interactions from a wide selection of TFs

We next aimed to further compare the network structures generated by LiPLike and the community. Since LiPLike aims to only include edges that uniquely model an interaction, we first analysed the Pearson correlation of inferred regulators in the LiPLike and top community networks. For each inferred edge, we searched for the closest correlating TF and found the median of these correlations to be significantly higher for the community predictions than LiPLike for all tested datasets ($\text{median}_{\text{community}} - \text{median}_{\text{LiPLike}} = 0.086$, 0.106, 0.053 and Mann–Whitney $P < 10^{-17}$, 10^{-124} and 10^{-30}), for *S.aureus*, *E.coli* and *in silico*, respectively (Fig. 5A). Thus, LiPLike avoided identifying edges when two or more regulators correlated, which was not the case for the community predictions. It can be noted that the median Pearson correlation coefficient of the closest correlating TF in the three networks was significantly lower in the *in silico* generated network than in the experimentally generated ones

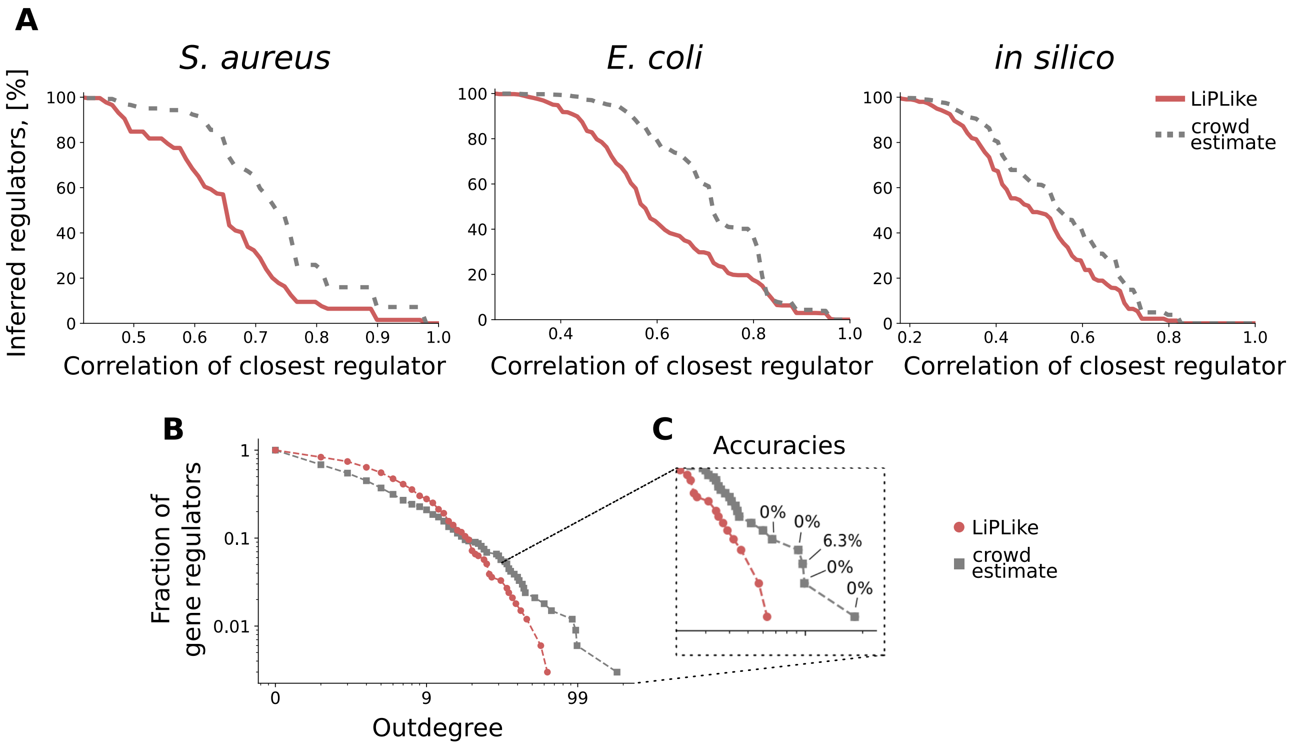

Fig. 5. Network properties. A) Cumulative distribution of the highest correlation with other regulators of putative interactions shown for LiPLike (red), and the crowd (grey) top-ranked interaction from respective DREAM5 networks. The regulators LiPLike identify tend to have on average fewer correlating regulators. For example, in the *E.coli* network, we observed a median Pearson correlation of $\rho = 0.57$. For the corresponding community prediction, 85.3% of all inferred regulators have a higher correlation than 0.57 to another regulator. This higher correlation indicates that LiPLike to a lesser degree predicts edges where there are several potential regulators to choose from. (**B**) Distribution of inferred edges for each transcription factor for LiPLike (red) and the community (grey). While the putative outdegrees of transcription factors in the community estimate appear to follow power law (as indicated by the straight line in the log-scale), LiPLike appears to select edges with a broader distribution profile. (**C**) Accuracies for the inferred top regulators in the community prediction were found to be low. The top regulators in the LiPLike network had similar accuracies to the overall LiPLike accuracy. (Color version of this figure is available at *Bioinformatics* online.)

($\text{median}_{S.aureus} = 0.61$, $\text{median}_{E.coli} = 0.54$ and $\text{median}_{in\ silico} = 0.39$ and Mann–Whitney $P_{in\ silico-S.aureus} < 10^{-21}$ and $P_{in\ silico-E.coli} < 10^{-19}$). It is possible that the lower Pearson correlations in the *in silico* data explain why LiPLike does not achieve a higher accuracy than the DREAM5 participants, as it is thus unlikely that two TFs can independently predict the expression of a gene.

Apart from predicting individual gene interactions, GRN inference can also be used to extract topological information on a biological system. For instance, network regulators with a disproportionately large outdegree have previously been used in predictive medicine (Gustafsson *et al.*, 2015). We hypothesized that such genes are more likely to have closely correlated regulators, and that LiPLike would therefore fail to identify their targets. Studying the *E.coli* network, we found that LiPLike indeed predicted networks containing fewer high-outdegree TFs than the community predictions. Whereas the community-predicted network followed power–law distribution on TF outdegrees, LiPLike produced a topology that was similar to an exponential distribution (Fig. 5B). In other words, the DREAM5 community produced a network where, compared to LiPLike, the predicted edges generally congregated around a few TFs. We then tested whether these TFs were important regulators that LiPLike failed to capture. Surprisingly, we observed accuracies much lower than that of the overall community accuracy (0.01 compared to 0.05 overall accuracy), with four of the five top TFs having no predicted edges with support in the gold standard (Fig. 5C). As a comparison, the top five TFs predicted by LiPLike showed a combined accuracy of 0.09, in line with the 0.10 overall LiPLike accuracy, indicating that LiPLike can robustly predict edges across node outdegrees.

Lastly, we tested whether the top regulators in the different DREAM5 participants overall had a low accuracy and continued to analyse all participants' regulators with a larger outdegree than any of the respective LiPLike regulators. Notably, the *S.aureus* and *E.coli* networks on average had lower accuracies than the overall accuracy of the method [*S.aureus* and *E.coli*: 88 of 90 and 158 of 200 hubs having lower accuracies than the method overall performance (binomial $P_{S.aureus} < 10^{-23}, P_{E.coli} < 10^{-16}$)], indicating that reduced accuracy in inferred edges from top gene regulators is a general property of GRN inference methods.

4 Discussion

The creation of GRNs is a pivotal tool for understanding gene expression patterns. However, such inferences have long struggled with low accuracy in predictions, to a large extent due to correlating expression profiles of potential regulators. In this article, we have presented the LiPLike, a novel method and software for high accuracy gene regulatory predictions from large biological datasets. As input, LiPLike takes data for dependent and independent variables and searches for cases where an independent variable uniquely explains the behaviour of a dependent variable. We showed that LiPLike could successfully infer gene–gene interactions from biological data by benchmarking it against the DREAM5 network inference challenge (Marbach *et al.*, 2012), achieving accuracies higher than all 36 DREAM5 participants for GRNs of *S.aureus* and *E.coli*. We also reported that combining LiPLike with networks from other GRN prediction methods significantly increased the accuracy for gene–gene interaction predictions, indicating that LiPLike can be used to remove false positive identifications from GRN predictions of other network inference methods.

LiPLike is related to the method profile likelihood, which aims to estimate the intervals that an estimated parameter can take with a retained fit to data (Cole *et al.*, 2014). However, LiPLike differs from the profile likelihood in three key aspects. First, the profile likelihood is most commonly applied to non-linear ordinary differential equation systems, while LiPLike assumes linear relations between independent and dependent variables, which increases the computation speed by several orders of magnitude. This increase in computational performance arises from the fact that the linear relationship is algebraically solvable in the form of the least-squares method, or variants thereof. It is important to point out that whereas a profile likelihood analysis can deal with almost any mathematical model, the restriction of LiPLike to linear models limits what biological dependencies can be captured. Nevertheless, the assumption that biological data come from a linear system is one of the most common approaches in GRN inference, and not exclusive to LiPLike. Second, LiPLike differs from the profile likelihood in that the profile likelihood tests which values a model parameter, β, can take with retained fit to data. LiPLike only tests the parameter β for the values minimizing the residual sum of squares and $\beta = 0$. Third, the profile likelihood estimates the uncertainty of a parameter, while LiPLike ranks all potential interactions in a network and interprets the increase in residual sum of squares from the two cases as an indication of the presence of an interaction.

While LiPLike is different from the profile likelihood in that the profile likelihood does not aim to infer edges in a network, there are alternative methods available for high accuracy GRN inference. Several methods have aimed to filter out gene–gene interaction identifications that are by-products of indirect correlations, for instance by studying the inverse covariance matrix of gene expression (Yuan, 2010). However, there are cases where this approach would fail. Consider a system of four genes, A–D. If C and D both are regulated by an interplay between A and B, C and D would arguably correlate more with each other than with A or B. Thus, a method that firmly avoids predictions where there is more than one possible model is needed to not make false positive identification. The RNI method (Nordling, 2013), for example, aims to only include edges that cannot be rejected by any model (Tjärnberg *et al.*, 2015). However, RNI might be too stringent and has been found not to make any interaction predictions from *in silico* generated data with SNR values commonly reported in biology (Tjärnberg *et al.*, 2015; Venet *et al.*, 2012). Another method aiming to address the problem of correlating explanatory variables is the random LASSO (Wang *et al.*, 2011), which bootstraps the explanatory variables in a series of steps and predicts a network by taking the average results from the bootstrap outcome. Thus, correlated explanatory variables will be predicted a smaller number of times, but will still be predicted in the final outcome (Wang *et al.*, 2011). An alternative approach to counter the problem of correlated explanatory variables is to cluster highly correlated variables into groups, as for example, done by the cMonkey algorithm, and use these groups as wider representations of explanatory variables (Reiss *et al.*, 2006). It should be noted, however, that such approaches capture broad changes in gene regulation, with identified interactions between clusters of genes as opposed to the single gene–gene interactions identified by LiPLike.

Moreover, a common approach to deal with false positive identifications is to include prior knowledge of the network that is being inferred. While LiPLike comes with an option to include prior matrices, this feature interferes with the rationale behind LiPLike, i.e. to only include interactions that cannot be replaced by a linear combination of all other regulators. Nevertheless, this rationale could be applied to local neighbourhoods of potential regulators too, as opposed to all genes.

LiPLike is, at its core, a comparison between two linear regression models. In the first case, the ability of a model containing all possible regulators, e.g. TFs, to fit a dependent variable is tested. In the second case, one regulator is removed, and the new fit of the model is calculated. If the decrease in fit is large, the removed variable had a unique explanatory power over the target gene that was being studied, and LiPLike would, therefore, identify and interaction.

Arguably, gene expression data often contain several genes that are regulated by the same biological processes, thereby correlate. This correlation impedes the accuracy of algorithms that aim to reverse-engineer the underlying structures of the gene regulation, as it is hard to distinguish between the directly and indirectly correlated variables when selecting variables to explain a gene, and false positive interactions are abundant. However, correlation between gene expression regulators is not the only factor that could invoke unidentifiable interactions in data. For example, gene expression is in reality regulated by the proteins of TFs, and most models use mRNA of TFs as a proxy of protein levels. In spite of that, gene expression data are known to be poorly correlated with the corresponding protein abundance (Fortelny *et al.*, 2017). In other words, regulators where mRNA expression is a poor proxy for the ability to control other genes will be hard to include in an mRNA-mRNA model. Nevertheless, there are today established approaches to estimate the activity of regulators from targets, such as discussed in Arrieta-Ortiz *et al.* (2015), and such approaches can indeed be used in combination with LiPLike.

The rationale behind LiPLike is to only identify edges that cannot be replaced by other interactions in the data. This approach is different from other approaches, which often try to infer the most probable network. This difference makes LiPLike highly stringent, and it is, therefore, closer to be a method for edge identification than a tool for full network reverse-engineering. Such identifications are important, for example, when planning costly follow-up experiments. Here, we showed that LiPLike seems to have a higher accuracy than other available tools for edge identification when explanatory variables are highly correlated. We further hypothesized this performance to stem from the properties of common GRN inference methods. Indeed, when encountering correlated independent variables, GRN-inference tools have been known to identify a regulator at random, or to include all potential regulators (Zou and Hastie, 2005). LiPLike, however, would identify none as a potential regulator. Importantly, we have also shown LiPLike to be able to remove false predictions from GRN produced from other methods, a property that can be used by anyone that wishes to stratify their predictions into sets of high and low confidence.

5 Conclusion

The occurrence of correlating explanatory variables poses a major obstacle when inferring GRNs. Available methods for GRN inference normally handle correlations by including several or one of the correlating explanatory variables. In this study, we present LiPLike, which identifies no interactions that cannot be uniquely inferred from data, and we show LiPLike to predict edges with higher accuracy than other state-of-the-art GRN inference tools in the DREAM5-challenge. Importantly, we also show that LiPLike can be used to remove false interactions from other methods, with the average increase of accuracies being 0.05, 0.04 and 0.36 to 0.47, 0.15 and 0.68, for *S.aureus*, *E.coli*, and *in silico*, respectively, and we recommend LiPLike be used on top of GRN estimations to give reliable predictions. In summary, we herein make gene interaction identification of high accuracy available for the community, using LiPLike together with other algorithms or as a stand-alone feature selection tool.

Acknowledgements

The authors would like to thank Andreas Tjärnberg for his input on *in silico* generated networks and David Martínez for proofreading the manuscript.

Funding

This work was supported by the Center for Industrial IT (CENIIT), the Swedish Research Council (2015-03807), and the Åke Viberg foundation.

Conflict of Interest: none declared.

References

Aghdam,R. *et al.* (2015) CN: a consensus algorithm for inferring gene regulatory networks using the SORDER algorithm and conditional mutual information test. *Mol. Biosyst.*, **11**, 942–949.

Alvarez,M.J. *et al.* (2018) A precision oncology approach to the pharmacological targeting of mechanistic dependencies in neuroendocrine tumors. *Nat. Genet.*, **50**, 979–989.

Arrieta-Ortiz,M.L. *et al.* (2015) An experimentally supported model of the *Bacillus subtilis* global transcriptional regulatory network. *Mol. Syst. Biol.,* **11**, 839.

Barzel,B. and Barabási,A.L. (2013) Network link prediction by global silencing of indirect correlations. *Nat. Biotechnol.*, **31**, 720–725.

Bonneau,R. *et al.* (2006) The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets *de novo*. *Genome Biol.*, **7**, R36.

Cheng,F. *et al.* (2018) Network-based approach to prediction and population-based validation of *in silico* drug repurposing. *Nat. Commun.*, **9**, 2691.

Cole,S.R. *et al.* (2014) Maximum likelihood, profile likelihood, and penalized likelihood: a primer. *Am. J. Epidemiol.*, **179**, 252–260.

Feizi,S. *et al.* (2013) Network deconvolution as a general method to distinguish direct dependencies in networks. *Nat. Biotechnol.*, **31**, 726–733.

Fortelny,N. *et al.* (2017) Can we predict protein from mRNA levels? *Nature*, **547**, E19–E20.

Friedman,N *et al.* (2000). Using Bayesian networks to analyze expression data. In: *Proceedings of the Fourth Annual International Conference on Computational Molecular Biology- RECOMB'. 00.* Association for Computing Machinery, pp. 127–135.

Guney,E. *et al.* (2016) Network-based *in silico* drug efficacy screening. Nat. Commun., **7**, 10331.

Gustafsson,M. *et al.* (2015) A validated gene regulatory network and GWAS identifies early regulators of T cell-associated diseases. *Sci. Transl. Med.*, **7**, 313ra178.

Kreutz,C. *et al.* (2013) Profile likelihood in systems biology. *FEBS J.,* **280**, 2564–2571.

Kwon,Y.K. (2016) Properties of Boolean dynamics by node classification using feedback loops in a network. *BMC Syst. Biol.*, **10**, 83.

Lefebvre,C. *et al.* (2010) A human B-cell interactome identifies MYB and FOXM1 as master regulators of proliferation in germinal centers. *Mol. Syst. Biol.*, **6**, 377.

Madar,A. *et al.* (2009) The Inferelator 2.0: a scalable framework for reconstruction of dynamic regulatory network models. In: *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society: Engineering the Future of Biomedicine, EMBC 2009. Conf. Proc. IEEE. Eng. Med. Biol. Soc.*, pp. 5448–5451.

Madar,A. *et al.* (2010) DREAM3: network inference using dynamic context likelihood of relatedness and the Inferelator. *PLoS One*, **5**, e9803.

Madhamshettiwar,P.B. *et al.* (2012) Gene regulatory network inference: evaluation and application to ovarian cancer allows the prioritization of drug targets. *Genome Med.*, **4**, 41.

Magnusson,R. *et al.* (2017) LASSIM-A network inference toolbox for genome-wide mechanistic modeling. *PLoS Comput. Biol.*, **13**, e1005608–19.

Marbach,D. *et al.*; The DREAM5 Consortium. (2012) Wisdom of crowds for robust gene network inference. *Nat. Methods*, **9**, 796–804.

Margolin,A.A. *et al.* (2006) ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, **7**, S7.

Morgan,D. *et al.* (2018) A generalized framework for controlling FDR in gene regulatory network inference. *Bioinformatics*, 1026–1032.

Noh,H. *et al.* (2018) Network perturbation analysis of gene transcriptional profiles reveals protein targets and mechanism of action of drugs and influenza A viral infection. *Nucleic Acids Res.*, **46**, e34.

Nordling,T.E.M. (2013) Robust inference of gene regulatory networks: System properties, variable selection, subnetworks, and design of experiments. Ph.D. Thesis, KTH Royal Institute of Technology, Stockholm, Sweden.

Prill,R.J. *et al.* (2010) Towards a rigorous assessment of systems biology models: the DREAM3 challenges. *PloS One*, **5**, e9202.

Reiss,D.J. *et al.* (2006) Integrated biclustering of heterogeneous genome-wide datasets for the inference of global regulatory networks. *BMC Bioinformatics*, **7**, 280.

Santolini,M. and Barabási,A.-L. (2018) Predicting perturbation patterns from the topology of biological networks. *Proc. Natl. Acad. Sci. USA*, **115**, E6375–E6383.

Tibshirani,R. (1991) Regression shrinkage and selection via the Lasso. *J. R. Stat. Soc. Series B Methodol.*, **58**, 267–288.

Tjärnberg,A. *et al.* (2015) Avoiding pitfalls in L1-regularised inference of gene networks. *Mol. Biosyst.*, **11**, 287–296.

Tjärnberg,A. *et al.* (2017) GeneSPIDER-gene regulatory network inference benchmarking with controlled network and data properties. *Mol. Biosyst.*, **13**, 1304–1312.

Venet,D. *et al.* (2012) A measure of the signal-to-noise ratio of microarray samples and studies using gene correlations. *PLoS One*, **7**, e51013.

Wang,C. *et al.* (2019) Efficient proximal gradient algorithm for inference of differential gene networks. *BMC Bioinformatics.*, **20**, 224.

Wang,S. *et al.* (2011) Random Lasso. *Ann. Appl. Stat.*, **5**, 468.

Xiong,J. and Zhou,T. (2012) Gene regulatory network inference from multifactorial perturbation data using both regression and correlation analyses. *PLoS One*, **7**, e43819.

Xu,H. *et al.* (2012) Sparse algorithms are not stable: a no-free-lunch theorem. *IEEE Trans. Pattern Anal. Mach. Intell*, **34**, 187–193.

Yuan,M. (2010) High dimensional inverse covariance matrix estimation via linear programming. *J. Mach. Learn. Res.*, **11**, 2261–2286.

Zhang,X. *et al.* (2015) Conditional mutual inclusive information enables accurate quantification of associations in gene regulatory networks. *Nucleic Acids Res.*, **43**, e31.

Zhao,P. and Yu,B. (2006) On model selection consistency of Lasso. **7**, 2541–2563.

Zou,H. and Hastie,T. (2005) Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Series B Stat. Methodol*, **67**, 301–320.