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; Madhamshettwar 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; Tjanberg 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 common approach is LiPLike (Kreutz, 2016), artificial neural networks (Marbach et al., 2012), 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 residual sum of squares and the point where 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 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 when 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://github.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.

\[ Y = f(X, \beta); \ Y, X, \beta \in \mathbb{R} \]  

(1)

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

\[ -LL(Y - f(X, \beta_{\hat{\beta}})) < T \]  

(2)

In Equation (2), the log-likelihood function, \( LL \), is studied with respect to one parameter \( \beta_i \). By varying \( \zeta \), a confidence interval can be defined as the values of \( \zeta \) for which the negative log-likelihood is below a threshold T. In other words, the profile likelihood tests which value a parameter \( \beta_i \) 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, \( \beta \).

\[ Y_{ij} = \sum_{k} \beta_{ik} X_{ik} + \epsilon \]  

(3)

In the annotations of Equation (3), \( Y_{ij} \) is a scalar corresponding to the expression of gene \( j \) at observation \( k \). Likewise, \( X_{ik} \) is a scalar corresponding to regulator \( i \) at observation \( k \). As is widely known, for overdetermined systems the vector \( \beta \) can be analytically estimated, via the methods of least squares, to minimize the sum of squared residuals. Now, for each parameter \( \beta_{ij} \), there are two parameter values that are of interest, namely the one minimizing the residual sum of squares and the point where \( \beta_{ij} = 0 \). Thus, we can quantify the relationship between these two points and introduce the term \( q_{ij} \).
Furthermore, if \( X \) exists, whereof two (Fig. 1B), in practice, Equation (4) compares the residual sum of squares of two ordinary least-squares regression problems; one with \( Y \) and \( Y \) and there will be a large increase in the residual sum of squares between the best fit and the baseline, 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 baseline, 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 baseline. As written in Equation (4), there are three properties that should be noted. First, if the system is not overdetermined, i.e. if \( \text{rank}(X) = n \), the denominator of Equation (4) equals zero, and thus \( q \) is undefined. Second, when computing \( q \), the residual sum of squares of the special case of \( b_j = 0 \) is divided by 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 \( q \) depends only on data properties and there are no additional hyperparameters that affect the performance of LiPLike.

### Algorithm 1. LiPLike pseudo-code

**Data:** matrices of dependent variables \( Y \) and independent variables \( X \)

**Result:** Scalar \( q \) indicating uniquely defined interaction

```plaintext
for every dependent variable \( j \) do
    for every independent variable \( i \) do
        \( q_{ij} = (j\text{-th residual sum of squares of full network except independent variable } i)/(j\text{-th residual sum of squares of full model}); \)
    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 \( q_{ij} \) 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, \( X_2 \) and \( X_3 \), the constraint \( b_{ij} = 0 \) will not independently give a considerable increase in the residual sum of squares. Thus, \( q_{ij} \approx 1 \), as \( b_{ij} \) will adjust accordingly. In Equation (4), there are three properties that should be noted. First, if the system is not overdetermined, i.e. if \( \text{rank}(X) = n \), the denominator of Equation (4) equals zero, and thus \( q \) is undefined. Second, when computing \( q \), the residual sum of squares of the special case of \( b_j = 0 \) is divided by 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 \( q \) 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 \( q_{ij} > 1 \), 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 \( q_j \) 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...
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 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 318 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 qi, larger than the max of the random q (S.aureus: 263 and in silico: 2203). To assess 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 subsession 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
removing false positive interactions seems to be a general property of LiPLike. By examining the interaction 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 predictions. Moreover, in the case of the biological networks, *S.aureus* and *E.coli*, LiPLike performs better than the community predictions, indicating that LiPLike identifies edges that the community failed to include.

community accuracies to increase from 9%, 5% to 53%, 15% and 70% (Fisher’s exact test $P < 1.8 \times 10^{-14}, 5.4 \times 10^{-18}, 2.6 \times 10^{-78}$ 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
LiPLike: high certainty gene–gene interaction predictions

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 $p = 0.57$. For the corresponding community prediction, 83.3% 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.

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 given probability. This method is used to calculate the probability of an observed data point given a model. In contrast, LiPLike uses a different approach to estimate the accuracy of gene–gene interactions. 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 restriction, the limitation 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, $\beta$, can take with retained fit to data. LiPLike only tests 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. H 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. This, 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 Arieta-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 Martinez for proofreading the manuscript.

## Funding

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

Conflict of Interest: none declared.

## References
