NARROMI: a noise and redundancy reduction technique improves accuracy of gene regulatory network inference

Xiujun Zhang1,2,3, Keqin Liu1,2,3, Zhi-Ping Liu4, Béatrice Duval3, Jean-Michel Richer3, Xing-Ming Zhao5,*, Jin-Kao Hao3,*, and Luonan Chen1,4,*

1Institute of Systems Biology, Shanghai University, Shanghai 200444, China, 2School of Communication and Information Engineering, Shanghai University, Shanghai 200072, China, 3LERIA, University of Angers, Angers 49045, France, 4Key Laboratory of Systems Biology, SIBS-Novo Nordisk Translational Research Centre for PreDiabetes, Shanghai Institutes for Biological Sciences, Chinese Academy of Sciences, Shanghai 200031, China and 5Department of Computer Science, School of Electronics and Information Engineering, Tongji University, Shanghai 201804, China

ABSTRACT
Motivation: Reconstruction of gene regulatory networks (GRNs) is of utmost interest to biologists and is vital for understanding the complex regulatory mechanisms within the cell. Despite various methods developed for reconstruction of GRNs from gene expression profiles, they are notorious for high false positive rate owing to the noise inherited in the data, especially for the dataset with a large number of genes but a small number of samples.

Results: In this work, we present a novel method, namely NARROMI, to improve the accuracy of GRN inference by combining ordinary differential equation-based recursive optimization (RO) and information theory-based mutual information (MI). In the proposed algorithm, the noisy regulations with low pairwise correlations are first removed by using MI, and the redundant regulations from indirect regulators are further excluded by RO to improve the accuracy of inferred GRNs. In particular, the RO step can help to determine regulatory directions without prior knowledge of regulators. The results on benchmark datasets from Dialogue for Reverse Engineering Assessments and Methods challenge and experimentally determined GRN of Escherichia coli show that NARROMI significantly outperforms other popular methods in terms of false positive rates and accuracy.

Availability: All the source data and code are available at: http://csb.shu.edu.cn/narromi.htm.

Contact: lichen@sibs.ac.cn, hao@info.univ-angers.fr and zhaoxingming@gmail.com.

Supplementary information: Supplementary data are available at Bioinformatics online.

Received on May 27, 2012; revised on September 18, 2012; accepted on October 14, 2012

1 INTRODUCTION
A major issue in systems biology is to construct and understand the gene regulatory networks (GRNs), which explicitly characterize regulatory processes in the cell (Basso et al., 2005). The development of high throughput technologies has produced tremendous amounts of gene expression data, which provide insights into the underlying regulatory mechanism of cellular machines (Hughes et al., 2000). The reconstruction or ‘reverse engineering’ of GRNs, which aims to dissect the underlying network of gene-gene interactions from the measurement of gene expression, is still a challenging task (Margolin et al., 2006a). For this reason, the Dialogue for Reverse Engineering Assessments and Methods (DREAM) project was established to encourage researchers to develop new efficient computation methods to infer robust GRNs (Marbach et al., 2010).

Recently, various approaches have been developed to infer GRNs from gene expression data with the motivation of improving the accuracy and scalability of network inference (Smet and Marchal, 2010). In general, these GRN inference methods fall into two categories, namely model-based approaches and machine learning-based approaches (Bansal et al., 2007; Li et al., 2011). For the model-based methods, chemical reaction of transcription and translation, as well as other cellular processes are described as linear or non-linear differential equations, in which the parameters represent the regulation strengths of the regulators. Representative algorithms in this category include multiple linear regression (Cantone et al., 2009; Gardner et al., 2003; Honkela et al., 2010; Tibshirani, 1996), singular value decomposition method (di Bernardo et al., 2005; Yeung et al., 2002), network component analysis (Chen et al., 2010; Liao et al., 2003) and linear programming (LP) (Wang et al., 2006). For the machine learning-based approaches, the network is inferred through measuring the dependences or causalities between transcriptional factors (TFs) and target genes (Kuffner et al., 2012). Popular methods in this category include partial correlation coefficient (De la Fuente et al., 2004, Saito et al., 2011), Bayesian network analysis (Li et al., 2011; Yeung et al., 2011), mutual information (MI) (Basso et al., 2005; Belcastro et al., 2011; Faith et al., 2007; Margolin et al., 2006b; Modì et al., 2011) and conditional mutual information (CMI) (Sumazín et al., 2011; Zhang et al., 2012).

As one of the most popular methods, MI has been widely used to construct GRNs because it provides a natural generalization of correlation owing to its capability of characterizing non-linear dependency (Basso et al., 2005). Furthermore, MI is able to deal with thousands of variables (genes) in the presence of a limited number of samples (Meyer et al., 2008). Despite these advantages, MI fails to distinguish indirect regulators from direct ones, i.e. it tends to overestimate the number of regulators.

*To whom correspondence should be addressed.
targeting the gene. In a GRN, the indirect regulations are the main source of false positives. Although some methods have been developed recently to remove these redundant indirect regulations, such as CMI (Frenzel and Pompe, 2007) and CMI-based path consistency algorithm (PCA-CMI) (Zhang et al., 2012), the high computational complexity makes them infeasible while calculating the high order MI. Another limitation of MI is that it only describes the correlation between two genes but is unable to determine the regulatory directions.

Different from the methods based on information theory, the model-based methods have the advantage of describing the regulatory dynamics and detecting the direction of regulations (Marbach et al., 2010). Furthermore, prior information, such as experimentally verified regulations, can be easily included in these models to improve the accuracy of network inference (Christley et al., 2009). Moreover, model-based methods are found useful to remove possible redundant indirect regulations by forcing sparseness on the model (Hurley et al., 2011; MacNeil and Walhout, 2011), such as the shortcut removing technique described as follows.

In this work, we propose a novel method, namely NARROMI, to improve the accuracy of GRN inference from gene expression data using a noise and redundancy (NAR)-reduction technology by combining ordinary differential equation (ODE)-based recursive optimization (RO) and information-theory based MI, and therefore it has the advantages of both model-based and machine learning-based methods. Specifically, the noisy regulations and partial significant indirect regulations can be firstly deleted and filtered using MI as a measure of dependence, and the redundant (indirect) regulations are subsequently removed gradually by the RO algorithm, thereby reducing both false positives and false negatives. In addition, our method can determine regulatory directions without prior information of regulators. The results on simulation datasets from DREAM challenge (Marbach et al., 2010) and experimentally confirmed network (Gama-Castro et al., 2011) in Escherichia coli with real gene expression data (Faith et al., 2008) show that our method significantly outperforms other popular methods in terms of false positives and accuracy.

2 METHODS

In general, the transcription process can be described by a mathematical model with differential equations based on mass action kinetics and Michaelis-Menten kinetics. However, the noise inherited in the data can decrease the performance of these models. Therefore, we present a new method NARROMI, which first reduces noisy regulations with MI and then uses RO technique to reduce redundant and indirect regulations gradually in the optimization model. The details of NARROMI can be found below.

2.1 Mathematical model of transcription procedure

During transcription, TF(s) binds to DNA sequences so as to recruit RNA polymerase II onto promoter region of DNA to initiate the transcription procedure (Sun et al., 2006; Wang et al., 2009), which can be described as follows.

\[
\text{DNA} + \beta_1 \cdot \text{TF}_1 + \cdots + \beta_c \cdot \text{TF}_c + \text{RPII} \xrightarrow{k_{+1}} \text{DNA} (\text{TF}_1)_{\beta_1} \cdots (\text{TF}_c)_{\beta_c} \text{RPII},
\]

where \(c\) is the number of TFs that are regulators of gene \(i\) \((i = 1, 2, \ldots, n)\), the stoichiometric coefficient \(\beta_j\), \(j = 1, 2, \ldots, c\), represents the effective abundance of TF\(j\) involved in the regulation of gene \(i\). DNA is the sequence of gene \(i\), and \(k_+, k_{-1}\), respectively, denotes the rate constant of forward reaction and reverse action. DNA(\text{TF}_1)_{\beta_1} \cdots (\text{TF}_c)_{\beta_c} \text{RPII}\) denotes the immobilized complex formed by TFs and RNA polymerase II. After transcription initiation, mRNAs are synthesized through the following irreversible reaction with rate constant \(k_2\)

\[
\text{DNA} (\text{TF}_1)_{\beta_1} \cdots (\text{TF}_c)_{\beta_c} \text{RPII} \xrightarrow{k_2} \text{mRNA}_i,
\]

At the translation level, mRNAs are translated into proteins

\[
\text{mRNA}_i \xrightarrow{k_{s1}} \text{Protein}_i,
\]

where \(k_s\), \(s = 1, 2, 3\), are the rate constants of reactions. According to the mass action law, the concentration changes of DNA(\text{TF}_1)_{\beta_1} \cdots (\text{TF}_c)_{\beta_c} \text{RPII}\) and mRNA can be described with following differential equations.

\[
d[\text{DNA} (\text{TF}_1)_{\beta_1} \cdots (\text{TF}_c)_{\beta_c} \text{RPII}] / dt = k_1 [\text{DNA} [\text{RPII}] \prod_{j=1}^{c} [\text{TF}_j]^{\beta_j} - k_{-1} + k_2] [\text{DNA} (\text{TF}_1)_{\beta_1} \cdots (\text{TF}_c)_{\beta_c} \text{RPII}],
\]

Assuming that (4) and (5) quickly reach an equilibrium state, i.e. \(d[\text{DNA} (\text{TF}_1)_{\beta_1} \cdots (\text{TF}_c)_{\beta_c} \text{RPII}] / dt = 0\) and \(d[\text{mRNA}_i] / dt = 0\), we can get \([\text{mRNA}_i] \propto \prod_{j=1}^{c} [\text{TF}_j]^{\beta_j} \). Let \(y(t) = [\text{mRNA}_i]\), \(A(t) = [\text{DNA} [\text{RPII}] \prod_{j=1}^{c} [\text{TF}_j]^{\beta_j} ]\), \(t = 1, 2, \ldots, m\), and \(y(t) \propto \prod_{j=1}^{c} A_j^{\beta_j}\). With a logarithm transformation, the above model can be described as a log-linear model \(\log(y(t)/y(0)) = \beta_1 \sum_{j=1}^{c} \log(A_j(t)/A_j(0))\). Let \(y_i(0) = \log(y(t)/y(0))\) and \(X_i(0) = \log(A(t)/A(0))\) where \(y_i(0)\) represents the expression level of gene \(i\) at time \(t\), and \(X_i(0)\) represents the activity of TF\(j\) at time \(t\), we get a linear model below by dropping \(i\) for simplicity.

\[
y_i = \beta X_i, i = 1, 2, \ldots, n,
\]

where \(y = (y_1, y_2, \ldots, y_n)\), \(X = (X_1, X_2, \ldots, X_c)\), \(X = (X_1)_{kx}^{n} \) is the number of target genes, \(c\) is the number of TFs and \(m\) is number of samples.

2.2 RO

For a target gene with expression level \(y\), we intend to define a regulation matrix \(\beta\) that fits well with the experimental data. \(\beta\) can be resolved by minimizing the error between inferred and observed expressions, i.e.

\[
\min_{\beta} \| y - \beta X \| + \lambda \| \beta \|,
\]

where \(X\) is the expression matrix of candidate TFs, and \(\lambda\) is a positive parameter that balances the error and sparse term in the objective function.

As TF activity can be approximated by the expression level of the gene encoding the TF, we suppose the gene expression level as the TF activity here. The model (6) is equivalent to

\[
\min_{\beta} \sum_{i=1}^{m} \| y_i - \sum_{j=1}^{c} \beta_j X_j \| + \lambda \sum_{j=1}^{c} \| \beta_j \|
\]

Let

\[
u_i + v_i = \| y_i - \sum_{j=1}^{c} \beta_j X_j \|, \quad u_i - v_i = y_i - \sum_{j=1}^{c} \beta_j X_j, \quad \xi_i + \eta_i = \| \beta_j \|, \quad \xi_i - \eta_i = \beta_j, \quad i = 1, 2, \ldots, m, j = 1, 2, \ldots, c,
\]
where \( u_i, v_i, \xi_j, \eta_j \geq 0 \). Then model (7) can be written as a standard LP model as follows.

\[
\begin{align*}
\min & \sum_{i=1}^{n} (u_i + v_i) + \lambda \sum_{j=1}^{p} (\xi_j + \eta_j) \\
\text{s.t.} & \quad u_i - v_i = y'_i - \sum_{j=1}^{p} (\xi_j - \eta_j)x'_j, \\
& \quad u_i, v_i, \xi_j, \eta_j \geq 0.
\end{align*}
\]

The above LP model (8) can be solved efficiently by any LP software such as GLPK LP/MIP solver (Wang et al., 2009). The parameter \( \lambda \) in model (8) is a positive value used to balance fitting and sparseness since GRNs are known to be sparse. The method of inferring GRNs simply by model (8) is called LP method in this article. LP is different from regression model-based LASSO, which reaches a least squares solution (Geeven et al., 2012).

Although the sparseness can be controlled by the parameter \( \lambda \) in model (8) to some extent, it is non-trivial to obtain an optimal network structure owing to the noise in the expression data. To ensure sparseness and reduce false positives, we set the variables with low regulation strengths to zero and re-estimate only the non-zero variables using model (8) in the next step. In this way, the accuracy of the network inferred by the second optimization step is improved with the strengthened sparsity by removing NAR regulations. The above procedure is repeated until there are no more non-zero variables. As this technique is composed of a series of optimization procedures, we call it RO. The technique to infer GRNs through solving model (8) recursively is called RO method in this article.

2.3 MI

The gene expression data can be described as vectors, in which the elements denote the expression values of genes under different conditions (samples). MI measuring the dependency between two genes \( X \) and \( Y \) can be defined as below (Altay and Emmert-Streib, 2010).

\[
I(X, Y) = - \sum_{x, y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}
\]

With the widely adopted hypothesis of Gaussian distribution for gene expression data, the formula (9) can be easily calculated using the following equivalent formula (Zhang et al., 2012).

\[
I(X, Y) = \frac{1}{2} \log \frac{|C(X)| \cdot |C(Y)|}{|C(X, Y)|},
\]

where \( C \) is the covariance matrix of variables, and \( |C| \) is the determinant of matrix \( C \). If genes \( X \) and \( Y \) are independent of each other, \( I(X, Y) = 0 \).

2.4 NARROMI algorithm

Figure 1 depicts the schematic view of our NARROMI method. The details are addressed as follows.
Algorithm (NARROMI)

Step 1: Noisy regulation detection
As is well known, the co-expressed genes are more possible to be regulated each other. For a target gene, the genes with high MI scores are co-expressed genes and more possible the true regulators. The MIs between all possible gene–gene (or regulator-target) pairs are firstly computed by formula (10). Given a threshold parameter \( \theta \) for deciding independence between variables, the regulations with MIs below the threshold are regarded as noisy regulations and removed from further analysis (Fig. 1A).

Note that in our algorithm, we select the TFs for each target gene, rather than select targets for each TF. In addition, those regulations with very large MIs will be kept as putative real regulations and integrated into the final GRN. If TFs are known in advance, the possible regulations of genes by these TFs are inferred. Otherwise, all genes are regarded as possible regulators, and all possible regulations are detected.

Step 2: Redundant regulation elimination
In this step, only the candidate regulators selected from Step 1 are used in the subsequent optimization for further inference of the network structure. In this way, the redundant regulators will be removed from the candidate regulators gradually using the RO algorithm for each gene. Usually, a gene is targeted by more than one TF. The regression model in RO can detect these combinatorial regulations simultaneously by selecting those regulators with high coefficients, while the regulators with low regulatory coefficients are eliminated.

Figure 1B gives the overview of the above procedure of RO, where nodes represent target or regulator genes, and arrows represent regulations from regulators to target genes. In more detail, the regulations are divided into three classes: non-regulations, indirect regulations and direct (true) regulations. The first two classes are redundant regulations, which can be removed by RO algorithm, and the real regulations will be kept until algorithm finished. Take target \( G_i \) as an example, edge \( G_5 \rightarrow G_i \) is a non-regulation, and edges \( R_3 \rightarrow G_i, R_4 \rightarrow G_i \) are two indirect regulations. All of them are redundant and will be removed by RO algorithm. Finally, edges \( G_1 \rightarrow G_i \) and \( G_2 \rightarrow G_i \) are kept as the two potential direct regulations of target \( G_i \).

Step 3: Integration of networks
To combine the linear and non-linear correlations between regulators and targets, the regulatory strengths inferred from RO algorithm, and the MI correlations are integrated in a linear combination way as follows.

\[
\beta = \text{sign}(\rho_{\text{MI}})\omega + (1 - \omega)\rho_{\text{RO}},
\]

where \( \rho_{\text{MI}} \) is the MI correlation, which is positive, \( \rho_{\text{RO}} \) is the regulatory strength (positive or negative) inferred by RO algorithm, \( \text{sign}(\rho_{\text{RO}}) \) is the sign \((\pm)\) of \( \rho_{\text{RO}} \), \( |\rho_{\text{RO}}| \) is the absolute of \( \rho_{\text{RO}} \) and parameter \( \omega \) is the weighting coefficient for MI and RO (Fig. 1C). The final regulatory strength is decided by the weight parameter \( \beta \), and the network topology is then determined (Treviño et al., 2012) (Supplementary Material).

3 RESULTS

To validate our method, NARROMI was applied to several simulation datasets and a real gene expression dataset. As for simulation data, the method was tested on the simulated benchmark GRNs with synthetic linear expression data and the widely used reference network in Yeast with synthetic non-linear expression data from DREAM3 challenge (Marbach et al., 2010). As for real gene expression data, we applied our method to the experiment confirmed network (Gama-Castro et al., 2011) in E. coli with real gene expression data (Faith et al., 2008).

The predictive results were evaluated by following measures, i.e. sensitivity or true positive rate (TPR), false positive rate (FPR), positive predictive value (PPV), accuracy (ACC) and Matthews Coefficient Constant (MCC). Mathematically, they are defined as

\[
\text{TPR} = TP/(TP + FN),
\]

\[
\text{FPR} = FP/(FP + TN),
\]

\[
\text{PPV} = TP/(TP + FP),
\]

\[
\text{ACC} = (TP + TN)/(TP + FP + TN + FN),
\]

\[
\text{MCC} = \frac{(TP \cdot TN - FP \cdot FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}},
\]

where TP, FP, TN and FN are the numbers of true positives, false positives, true negatives and false negatives, respectively.

To evaluate the performance of NARROMI, we compared it with several popular methods including LP, RO, regression model-based LASSO (Geeven et al., 2012), MI-based ARACNE (Margolin et al., 2006b) and random forest-based GENIE3 (Huynh-Thu et al., 2010), where the two alternatives with parameters ‘sign’ and ‘all’ in GENIE3 were considered here, as they performed best in the DREAM challenges (Supplementary Material).

For example, the regularization parameter \( \lambda \) of methods LP, RO and NARROMI was set to 1; the ensemble parameter of method GENIE3 was set to 1000; the threshold of MI filtering in method NARROMI was set to 0.05.

3.1 Evaluation on simulation data

In this section, we show computational results of our method on two types of simulation datasets, i.e. linear and non-linear expression dataset. For linear expression data, we generated the benchmark networks and related expression datasets using the simple conventional method in statistics. For non-linear expression data, we performed our method on the widely used reference network in Yeast with synthetic non-linear expression data from DREAM3 challenge (Marbach et al., 2010).

3.1.1 Artificial linear expression data For linear expression data, the benchmark networks and corresponding gene expression data were generated according to the network size, degree of sparseness and rate of noise. First, the benchmark network was generated according to the degree of network, number of target genes and number of candidate regulators. Second, the expression data of the regulators were generated randomly using the Gaussian distribution function. Third, the expression data of the target genes were generated by linear combination with the expression data of regulators. Last, the Gaussian noise was added to the expression data of target genes. In the experiments, we generated different networks with 10, 100, 500, 1000 and 5000 regulators and expression datasets under 5, 10, 15, 20 and 25 samples, respectively (Table 1). The average input degree for each target was set to ~2, and the noise rate was set to 10%.

Figure 2 shows the ROC curves by different methods on datasets of sizes 10, 100 and 1000. For the networks with sizes 500 and 5000, the ROC curves are showed in...
Supplementary Figure S1. From the figures, we can clearly see that the performance of our NARROMI method is superior to other methods with AUC score $>0.90$. Table 1 summarizes the results obtained by different methods with respect to distinct performance indices. From Table 1, we can see that RO performs better than LP and LASSO, both ARACNE and GENIE3 have good performance on large-scale networks and our method NARROMI performs best with the highest AUC values of 0.992, 0.928 and 0.937 on all three datasets. The results show that NARROMI is more robust than other methods on different network sizes. When the network size is large enough with thousands of genes, the accuracy of NARROMI is still high enough, whereas the large network size degrades the performance of other methods significantly. For the networks with sizes 500 and 5000, the results with respect to other performance indexes are given in Supplementary Tables S1 and S2, where NARROMI performs best.

3.1.2 Artificial non-linear expression data
For non-linear expression data, the widely used benchmark networks with expression datasets from DREAM challenge were adopted here to evaluate our method. The gold standard networks were generated with the non-linear ODE systems in which the network structures were determined with detailed dynamics of both transcriptional and translational processes (Schaffter et al., 2011). In this work, the DREAM3 datasets about Yeast knock-out genes with sizes 10 and 50 were used (Marbach et al., 2010).

Firstly, NARROMI was applied to the Yeast gene expression data with network sizes 10 and 10 samples. Figure 3 shows the structure of our inferred network and the ROC curves obtained by different methods. Figure 3A shows the true network with 10 genes and 10 edges, and Figure 3B shows the network inferred by NARROMI. From the figure, we can see that most edges were recovered by NARROMI, although some regulations were missed, such as G6-G4 and G9-G4 with dashed lines. In addition, five of eight inferred edges (62.5%) were detected correctly with respect to regulatory directions. This indicates that NARROMI can detect most regulatory directions without the information of TFs. The comparison of NARROMI with other methods was shown in Figure 3C, where NARROMI outperforms other methods significantly with an AUC score of 0.938.

### Table 1. Comparison on networks with sizes 10, 100 and 1000

<table>
<thead>
<tr>
<th>Method</th>
<th>TPR</th>
<th>FPR</th>
<th>PPV</th>
<th>ACC</th>
<th>MCC</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size 10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LASSO</td>
<td>0.667</td>
<td>0.122</td>
<td>0.546</td>
<td>0.840</td>
<td>0.505</td>
<td>0.776</td>
</tr>
<tr>
<td>LP</td>
<td>0.778</td>
<td>0.024</td>
<td>0.789</td>
<td>0.859</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RO</td>
<td>0.778</td>
<td>0.073</td>
<td>0.900</td>
<td>0.677</td>
<td>0.886</td>
<td></td>
</tr>
<tr>
<td>ARACNE</td>
<td>0.556</td>
<td>0.146</td>
<td>0.456</td>
<td>0.800</td>
<td>0.380</td>
<td>0.748</td>
</tr>
<tr>
<td>GENIE3 FR_sqrt</td>
<td>0.444</td>
<td>0.171</td>
<td>0.364</td>
<td>0.760</td>
<td>0.254</td>
<td>0.748</td>
</tr>
<tr>
<td>GENIE3 FR_all</td>
<td>0.333</td>
<td>0.073</td>
<td>0.500</td>
<td>0.820</td>
<td>0.308</td>
<td>0.753</td>
</tr>
<tr>
<td>NARROMI</td>
<td>1.000</td>
<td>0.024</td>
<td>0.900</td>
<td>0.980</td>
<td>0.937</td>
<td>0.992</td>
</tr>
<tr>
<td>Size 100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LASSO</td>
<td>0.474</td>
<td>0.010</td>
<td>0.474</td>
<td>0.980</td>
<td>0.464</td>
<td>0.770</td>
</tr>
<tr>
<td>LP</td>
<td>0.474</td>
<td>0.010</td>
<td>0.474</td>
<td>0.980</td>
<td>0.464</td>
<td>0.817</td>
</tr>
<tr>
<td>RO</td>
<td>0.474</td>
<td>0.010</td>
<td>0.474</td>
<td>0.980</td>
<td>0.464</td>
<td>0.827</td>
</tr>
<tr>
<td>ARACNE</td>
<td>0.421</td>
<td>0.042</td>
<td>0.163</td>
<td>0.948</td>
<td>0.239</td>
<td>0.887</td>
</tr>
<tr>
<td>GENIE3 FR_sqrt</td>
<td>0.263</td>
<td>0.015</td>
<td>0.250</td>
<td>0.971</td>
<td>0.241</td>
<td>0.809</td>
</tr>
<tr>
<td>GENIE3 FR_all</td>
<td>0.263</td>
<td>0.018</td>
<td>0.217</td>
<td>0.968</td>
<td>0.223</td>
<td>0.778</td>
</tr>
<tr>
<td>NARROMI</td>
<td>0.526</td>
<td>0.008</td>
<td>0.556</td>
<td>0.983</td>
<td>0.532</td>
<td>0.928</td>
</tr>
<tr>
<td>Size 1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LASSO</td>
<td>0.631</td>
<td>0.0006</td>
<td>0.667</td>
<td>0.9987</td>
<td>0.648</td>
<td>0.839</td>
</tr>
<tr>
<td>LP</td>
<td>0.684</td>
<td>0.0008</td>
<td>0.619</td>
<td>0.9986</td>
<td>0.650</td>
<td>0.812</td>
</tr>
<tr>
<td>RO</td>
<td>0.684</td>
<td>0.0007</td>
<td>0.650</td>
<td>0.9987</td>
<td>0.666</td>
<td>0.867</td>
</tr>
<tr>
<td>ARACNE</td>
<td>0.211</td>
<td>0.0006</td>
<td>0.400</td>
<td>0.9979</td>
<td>0.289</td>
<td>0.929</td>
</tr>
<tr>
<td>GENIE3 FR_sqrt</td>
<td>0.316</td>
<td>0.0066</td>
<td>0.061</td>
<td>0.9895</td>
<td>0.136</td>
<td>0.888</td>
</tr>
<tr>
<td>GENIE3 FR_all</td>
<td>0.263</td>
<td>0.0092</td>
<td>0.070</td>
<td>0.9920</td>
<td>0.133</td>
<td>0.874</td>
</tr>
<tr>
<td>NARROMI</td>
<td>0.578</td>
<td>0.0007</td>
<td>0.650</td>
<td>0.9987</td>
<td>0.666</td>
<td>0.937</td>
</tr>
</tbody>
</table>

The best performer for the relative item is noted in bold.

LASSO, regression-based method; LP, linear programming-based method; RO, recursive optimization-based method; ARACNE, MI-based method; GENIE3, random forests-based methods; NARROMI, method based on RO and MI.

Fig. 2. ROC curves of several methods on networks with different sizes. The solid line with star points is the ROC curve of method NARROMI. The dashed lines with different points are ROC curves of method LASSO, LP, RO, ARACNE, GENIE3 FR sqrt and GENIE3 FR all, respectively. (A) The ROC curves on network with size 10. (B) The ROC curves on network with size 100. (C) The ROC curves on network with size 1000.
Secondly, the Yeast gene expression data with network size 50 was used to evaluate NARROMI and other methods. Table 2 shows the results obtained by different methods with respect to distinct performance indexes. The ROC curves by these methods can be found in Supplementary Figure S2. From the results, we can observe that NARROMI performs better than most methods except the method GENIE3 with parameter ‘sqrt’ for one case. This is also consistent with the analysis that no single inference method performs optimally across all datasets (Marbach et al., 2012).

To evaluate the effect of Step 2 (RO) on the performance of NARROMI, we compared the results of NARROMI with or without RO. The results can be found in Supplementary Tables S3 and S4, from which we can see that Step 2 of NARROMI can indeed remove most indirect regulations, and the false positives can be reduced significantly from 0.188 to 0.037 and from 0.109 to 0.032, respectively.

### 3.2 Identification of gene regulatory interactions in E. coli

Except the above simulation datasets, NARROMI was also applied to construct regulatory networks from real gene expression data. We evaluated our NARROMI on the experimentally verified reference network in E. coli (Gama-Castro et al., 2011). The expression data was drawn from the well known E. coli data bank (Faith et al., 2008). In the experimentally verified networks, there are 2675 edges between 160 regulators and 1258 targets that can be found in the expression dataset. For each target, there are \( \frac{C_2}{C_2} \) regulators on average, which is consistent with the setting for the regulation degrees of simulation dataset in Section 3.1.1.

To evaluate the performance of our method, the AUC scores were computed. Moreover, the number and proportion of TFs and target genes that were predicted correctly were also recorded. We do not compare NARROMI against GENIE3 with parameter ‘all’ that is time consuming in this case. From the results in Table 3, we can see that NARROMI performs better than other methods with the highest average AUC scores for both TFs and target genes of 0.754 and 0.735, respectively. Figure 4 and Supplementary Figure S3 show our inferred network structures, which indicate high overlap with the reference network.

In Figure 4A, the inferred target genes of TF AppY were shown, and those overlapped with the reference network were marked in gray. Figure 4B shows the inferred regulators of target gene gadB, and the regulators that were predicted correctly were marked in gray.

### Table 2. Comparison of different methods on networks with sizes 10 and 50 in DREAM3

<table>
<thead>
<tr>
<th>Method</th>
<th>TPR</th>
<th>FPR</th>
<th>PPV</th>
<th>ACC</th>
<th>MCC</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size 10</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LASSO</td>
<td>0.600</td>
<td>0.837</td>
<td>0.211</td>
<td>−0.191</td>
<td>0.703</td>
<td></td>
</tr>
<tr>
<td>LP</td>
<td>0.100</td>
<td>0.412</td>
<td>0.533</td>
<td>−0.202</td>
<td>0.738</td>
<td></td>
</tr>
<tr>
<td>RO</td>
<td>0.000</td>
<td>0.296</td>
<td>0.456</td>
<td>−0.252</td>
<td>0.798</td>
<td></td>
</tr>
<tr>
<td>ARACNE</td>
<td>0.900</td>
<td>0.112</td>
<td>0.500</td>
<td>0.676</td>
<td>0.618</td>
<td>0.930</td>
</tr>
<tr>
<td>GENIE3_FR_sqrt</td>
<td>0.700</td>
<td>0.112</td>
<td>0.437</td>
<td>0.867</td>
<td>0.483</td>
<td>0.919</td>
</tr>
<tr>
<td>GENIE3_FR_all</td>
<td>0.700</td>
<td>0.138</td>
<td>0.389</td>
<td>0.844</td>
<td>0.442</td>
<td>0.894</td>
</tr>
<tr>
<td>NARROMI</td>
<td>0.700</td>
<td>0.050</td>
<td>0.636</td>
<td>0.922</td>
<td>0.623</td>
<td>0.938</td>
</tr>
<tr>
<td><strong>Size 50</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LASSO</td>
<td>0.351</td>
<td>0.129</td>
<td>0.855</td>
<td>0.113</td>
<td>0.711</td>
<td></td>
</tr>
<tr>
<td>LP</td>
<td>0.389</td>
<td>0.085</td>
<td>0.130</td>
<td>0.899</td>
<td>0.182</td>
<td>0.669</td>
</tr>
<tr>
<td>RO</td>
<td>0.494</td>
<td>0.131</td>
<td>0.109</td>
<td>0.857</td>
<td>0.181</td>
<td>0.727</td>
</tr>
<tr>
<td>ARACNE</td>
<td>0.597</td>
<td>0.082</td>
<td>0.192</td>
<td>0.908</td>
<td>0.303</td>
<td>0.832</td>
</tr>
<tr>
<td>GENIE3_FR_sqrt</td>
<td>0.481</td>
<td>0.078</td>
<td>0.167</td>
<td>0.908</td>
<td>0.245</td>
<td>0.843</td>
</tr>
<tr>
<td>GENIE3_FR_all</td>
<td>0.442</td>
<td>0.073</td>
<td>0.164</td>
<td>0.912</td>
<td>0.231</td>
<td>0.796</td>
</tr>
<tr>
<td>NARROMI</td>
<td>0.532</td>
<td>0.062</td>
<td>0.217</td>
<td>0.925</td>
<td>0.307</td>
<td>0.839</td>
</tr>
</tbody>
</table>

The best performer for the relative item is noted in bold.

LASSO, regression-based method; LP, linear programming-based method; RO, recursive optimization-based method; ARACNE, MI-based method; GENIE3, random forests-based methods; NARROMI, method based on RO and MI.

respect to PPV, ACC, MCC and AUC are shown in Table 2, where NARROMI is superior to other methods.

To evaluate the performance of our method, the AUC scores were computed. Moreover, the number and proportion of TFs and target genes that were predicted correctly were also recorded. We do not compare NARROMI against GENIE3 with parameter ‘all’ that is time consuming in this case. From the results in Table 3, we can see that NARROMI performs better than other methods with the highest average AUC scores for both TFs and target genes of 0.754 and 0.735, respectively. Figure 4 and Supplementary Figure S3 show our inferred network structures, which indicate high overlap with the reference network. In Figure 4A, the inferred target genes of TF AppY were shown, and those overlapped with the reference network were marked in gray. Figure 4B shows the inferred regulators of target gene gadB, and the regulators that were predicted correctly were marked in gray.

Fig. 3. Comparison of a 10-gene network inferred from DREAM3 dataset. (A) The true network with 10 nodes and 10 edges. (B) The network inferred by NARROMI. The solid lines are true positive edges of the inferred network. The edges G4-G1, G5-G1 and G7-G10 with dotted lines are false positive edges, which are mistaken for the correlation from opposition, whereas the edges G4-G9 and G6-G4 with dashed lines are missed for the low causalities. (C) The comparison of ROC curves by method NARROMI with other methods. The solid line with star points is the ROC curve of method NARROMI, which superior to other methods.
### 4 DISCUSSION

In this article, we proposed a novel method NARROMI to infer GRNs from gene expression data by combining ODE-based RO and information theory-based MI. From results on the benchmark datasets, NARROMI is effective and outperforms other methods significantly. The good performance of NARROMI may be contributed by following factors.

First, it reduces the NAR regulation through two steps, i.e. MI filtering and RO sparseness. As the first step, MI removes the noisy regulations and thus provides clear preliminary structure to the following optimizations (RO). The subsequent RO procedure reduces the redundant (indirect) regulations gradually, thereby improving the performance of network inference.

Second, the regulatory networks inferred by NARROMI consist of both linear and non-linear correlations between regulators and targets. The RO technique detects the regulators for target genes using linear systems, which describe the chemical reactions of transcription and translation as linear differential equations. The integration with MI in the last step of the algorithm takes into account the non-linear correlations between regulators and targets, which makes NARROMI superior to general linear model based methods.

Despite the advantages of NARROMI, there is still room to improve it. For example, as a popular correlation measure, MI can identify most linear and non-linear correlations, but there are also some special non-linear correlations such as sinusoidal that MI cannot detect. Recently, a meaningful measurement, maximal information coefficient, has been proposed to detect associations from large dataset (Reshef et al., 2011), which may help to improve the performance of NARROMI. Although the RO step in NARROMI is able to determine regulatory directions, other popular techniques for causal regulations may obtain better results and will be considered in NARROMI in the future.

### 5 CONCLUSION

We proposed a novel method NARROMI to improve the accuracy of GRN inference by simultaneously implementing the NAR regulation reduction. In this algorithm, the noisy regulations with low pair-wise correlations and the redundant regulations from indirect regulators are removed with MI and RO, respectively. Moreover, the dimension shrinking of the technique improves the efficiency of optimization and further increases the accuracy of network inference. The method was validated on the simulated benchmark GRNs from DREAM challenge and the experimentally verified network in *E. coli* with real gene expression data. The cross-validation results confirmed the effectiveness of our method (NARROMI), which outperformed previous methods.

### ACKNOWLEDGEMENTS

The authors thank the anonymous reviewers for their constructive comments, which greatly help them improve their manuscripts. They also thank Xiaodong Zhang in their group for the technical help and Dr. Văn Anh Huynh-Thu of University of Liège, Belgium, for the help with the use of software GENIE3.

**Funding:** Sino-French Cai Yuanpei Program from CSC (20106050), NSFC (91029301, 61134013, 61072149, 91130032, 61103075, 31100949), Innovation Program of Shanghai Municipal Education Commission (13ZZ072), Shanghai Pujiang Program, Chief Scientist Program of SIBS from CAS (2009CSP002), Knowledge Innovation Program of SIBS of CAS (2011KIP203), the Knowledge Innovation Program of CAS (KSCX2-EW-R-01), National Center for Mathematics and Interdisciplinary Sciences of CAS, and the FIRST Program initiated by CSTP, Radapop and LigeRO Projects (2009-2013, Pays de la Loire Region, France).

**Conflict of Interest:** none declared.
REFERENCES


