BMC Genomics

RESEARCH Open Access

D3GRN: a data driven dynamic network construction method to infer gene regulatory networks

Xiang Chen[1], Min Li[1*], Ruiqing Zheng[1], Fang-Xiang Wu[2] and Jianxin Wang[1]

From 2018 International Conference on Intelligent Computing (ICIC 2018) and Intelligent Computing and Biomedical Informatics (ICBI) 2018 conference
Wuhan and Shanghai, China. 15-18 August 2018, 3–4 November 2018

Abstract

Background: To infer gene regulatory networks (GRNs) from gene-expression data is still a fundamental and challenging problem in systems biology. Several existing algorithms formulate GRNs inference as a regression problem and obtain the network with an ensemble strategy. Recent studies on data driven dynamic network construction provide us a new perspective to solve the regression problem.

Results: In this study, we propose a data driven dynamic network construction method to infer gene regulatory network (D3GRN), which transforms the regulatory relationship of each target gene into functional decomposition problem and solves each sub problem by using the Algorithm for Revealing Network Interactions (ARNI). To remedy the limitation of ARNI in constructing networks solely from the unit level, a bootstrapping and area based scoring method is taken to infer the final network. On DREAM4 and DREAM5 benchmark datasets, D3GRN performs competitively with the state-of-the-art algorithms in terms of AUPR.

Conclusions: We have proposed a novel data driven dynamic network construction method by combining ARNI with bootstrapping and area based scoring strategy. The proposed method performs well on the benchmark datasets, contributing as a competitive method to infer gene regulatory networks in a new perspective.

Keywords: Gene regulatory network, Dynamic network construction, Regression, DREAM challenge

Introduction

Gene regulation plays an important role in gene transcription [1, 2], gene differentiation [3], cell fate decisions [4, 5], complex diseases [6]. To elucidate the structure of gene regulatory networks (GRNs) has been a central effort of the interdisciplinary field of systems biology. With the advent of high-throughput technologies such as microarrays and RNA sequencing, tremendous amounts of data have been generated, which makes it feasible to infer GRNs from exclusive expression data or multiple classes of data based on computational methods [7]. However, inferring the GRN only from gene expression data remains a daunting task due to the small number of available measurements and the high dimensional, noisy data.

Various methods have been proposed for GRN inference [8–10], such as correlation and information theory based methods, Boolean Networks (BNs), Bayesian networks, ordinary differential equations (ODEs), and regression based methods. These approaches can be divided into two categories with different levels of granularity. The first category predicts the presence or absence of gene interactions to give a static network describing only the topological information, correlation and information theory based methods belong to this category. Other methods belong to the second category, which predicts the rate of gene interactions describing both topological and

*Correspondence: limin@mail.csu.edu.cn
[1] School of Computer Science and Engineering, Central South University, Changsha, China
Full list of author information is available at the end of the article

dynamic information. ODEs and regression based methods are two kinds of most widely applied techniques in all of the classes of GRN inference methods.

In correlation and information theory based methods, other than the simple Pearson correlation [11] one of the most favored metrics is mutual information (MI) [12], which is capable of capturing complex non-linear and non-monotonic dynamics between pairs or groups of genes [13, 14]. ARACNE [15] employs Data Processing Inequality (DPI) to discard indirect interactions from a triplet of genes. Subsequently based on the same purpose, conditional mutual information (CMI) [16], local overlapped gene clusters based conditional mutual information (Loc-PCA-CMI) [17], part mutual information (PMI) [18] and partial information decomposition (PID) [19] are proposed to eliminate false positive or indirect regulatory links as much as possible.

In BNs, the alternative states of a gene are represented with discrete value 0 (inactive) and 1 (active), the regulatory interactions are described by Boolean logic [20]. Probabilistic Boolean Networks (PBNs) [21] brings in probability into standard BNs to express uncertainty in the regulatory logic. Typical variants, such as Stochastic Boolean networks (SBNs) [22], aim to improve the computational performance of PBNs. BNs's weakness is that the models only consider genes in discrete states. Thus the detail information involved in real gene expressions can not be captured effectively.

Bayesian networks, including traditional Bayesian networks [23, 24] and dynamic Bayesian networks (DBNs) [25], model the gene regulation processes based on probability and graph theory. Bayesian networks regard regulations of genes as the dependence probabilities between random variables and learn the optimal structures from gene profiles. Bayesian networks suffer from considerable computational overheads, despite recent advances [26] hence are not applicable to large genome-wide data sets.

ODEs provide an infinitesimal description of the regulation dynamics [27], by relating the rate of change (time derivative) of a gene to its expression value. Inferelator [28], S-system model [29–31] are typical approaches in ODEs. Generally, ODEs based methods are flexible by taking advantage of large parameters space estimation. As a result, akin to Bayesian networks tremendous computation is required to fulfill the task.

Most regression based methods formalize the GRN inference problem as a feature selection problem and construct the GRN with some ensemble strategy. GENIE3 [32] is recognized as state-of-the-art on some benchmark datasets [33], which is based on feature selection with ensembles of random forests. TIGRESS [34] uses least angle regression (LARS) with stability selection combined to solve the GRN inference problem. The NIMEFI method [35] explores the potential of several ensemble methods, such as GENIE3, Ensemble Support Vector Regression (E-SVR) and Ensemble Elastic Net (E-EL) [36], and combines the predictions of these methods under a general framework. bLARS [37] can be viewed as a variant method of TIGRESS, in which regulation interactions are modeled from a predefined family of functions, and the final GRN is obtained by a modified LARS algorithm with bootstrapping.

Recently, data driven dynamic network construction especially in a physical system is a pretty attractive and interesting topic. SINDy [38] assumes that there are only a few important terms that govern the dynamics so that the equations are sparse in the space of possible functions. It then uses sparse regression to determine the fewest terms in the dynamic governing equations required to accurately represent the data. ARNI [39] is a model-independent framework for inferring direct interactions in network dynamical systems, which is relying only on their non-linear collective dynamics. It solves nonlinear systems of differential equations via functional decomposition and expansions in basis functions.

Though bLARS, SINDy and ARNI are proposed in different areas, they are somehow similar in the basic thought. Table 1 shows the comparison of these methods from three different aspects. Formal function decomposition means whether the method has a formal description with equations of function decomposition; sparse group constraints indicates whether the method utilizes sparse group constraints with the candidate terms, while network based construction indicates if the method aims to recover a whole network structure. Both SINDy and ARNI do not intend to address the problem of uncovering the physical mechanism from network level, but solely from the unit level instead. Motivated by the fact that none of the methods covers all the three points, in this study we propose a new data driven dynamic network construction method, contributing as the first attempt including above three aspects systematically. D3GRN casts the regulatory relationship of each target gene into functional decomposition problem and solves each sub problem in the way of feature selection, by using the Algorithm for Revealing Network Interactions (ARNI). The whole network structure is recovered by the bootstrapping strategy with the area based scoring method. We compare the performance of our method D3GRN to several state-of-the-art

Table 1 Comparison of the related methods

	bLARS	SINDy	ARNI	D3GRN
Formal function decomposition	×	✓	✓	✓
Sparse group constraints	✓	×	✓	✓
Network based construction	✓	×	×	✓

methods in DREAM4 and DREAM5 gene reconstruction challenge, and the results show our method performs competitively in terms of AUPR.

Method

Problem definition

GRNs can be viewed as directed acyclic graphs (DAGs) if both up-streaming or down-streaming regulatory relationships among genes are not considered and the self-regulatory mechanism is ignored. In a DAG, each node corresponds to a gene and each edge represents a regulatory relationship between genes. The same as many other ensemble methods (e.g., [32], [34, 35, 40–42]), which does not utilize the information of different experimental conditions (e.g, gene-knockouts, perturbations and even replicates), we use a general framework for GRNs inference problem only based on gene expression data. As the input gene expression data, we consider the measurements for N genes in M experimental conditions. The gene expression data A is thus defined as follows:

$$A = [x_1, x_2, \dots, x_N] \in \mathbb{R}^{M \times N} \tag{1}$$

where x_i is a column vector of expression values of the i-th gene in all the M experimental conditions.

GRN inference methods predict the regulatory links between genes from gene expression data A. Most methods provide a ranking list of the potential regulatory links from the most to the less confidence. Different DAGs can be subsequently obtained by selecting varying threshold values on this ranking list. As it is beneficial to the end-user to explore the network at all sorts of threshold levels [40], we focus only on the ranking issue in this study. Of note, the ranking is the standard prediction format of the "Dialogue for Reverse Engineering Assessments and Methods" (DREAM) [43] challenges, wherein various GRN inference methods have been proposed. Besides, we do not consider the stability of the obtained networks from the ranking.

In order to infer a regulatory network from the expression data A, we compute a weight score S_{ij} for a potential edge directed from gene i to gene j, where the edge indicates that gene i regulates gene j on expression level and the weight score S_{ij} represents the strength that gene i regulates (including both upstream regulates and downstream regulates) gene j.

Network inference with ensemble regression methods

Motivated by the success of ensemble methods based on feature selection (e.g., GENIE3 [32] and TIGRESS [34]), the GRN inference problem with N genes can be decomposed into N sub problems, where each sub problem can be viewed as a feature selection issue in machine learning [44]. More specifically, for each target gene, we wish to determine the subset of genes which directly influence it from the expression level. Let A is the gene expression data defined in Eq. (1), the i-th gene is the target gene, and we define other candidate regulators with expression values in M experimental conditions as:

$$x^{-i} = [x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_N] \tag{2}$$

and the feature selection problem can be defined as:

$$x_i = F(x^{-i}) + \epsilon, \forall i \in \{1, 2, \dots, N\} \tag{3}$$

where F is any smooth, typically nonlinear function of the expression in x^{-i} of genes that are directly connected to gene i, and ϵ is the noise term [32, 34]. By aggregating the N individual gene rankings, we can obtain a global ranking of all regulatory links in a GRN.

GRN inference with D3GRN

The pseudo code of D3GRN Algorithm 1 is given below. A_j refers to the jth column of the matrix A, and A_I is the submatrix that contains only the columns in the index set I of A. Suppose that the input gene expression data matrix $A \in \mathbb{R}^{M \times N}$ and the indices of the transcription factors $I \subset \{1, \dots, N\}$, as well as the bootstrapping numbers and ARNI steps L are given. Then for each target gene j, and bootstrapping run i, by resampling the expression matrix A with replacement, the respective values of the target gene j denoted as y, and the expression values of the remaining transcription factors X are obtained, respectively. The ARNI algorithm is invoked to return an ordered list of selected transcription factors denoted by SM_j. Finally, after all the b bootstrapping runs end, the matrix SM is passed to the area based scoring method that assigns a score between 0 and 1 to an edge between a transcription factor and a target gene. Bootstrapping

Algorithm 1 D3GRN Pseudo Code

```
Require: $A \in \mathbb{R}^{M \times N}, I \subset \{1, \dots, N\}, |I| = n$   ▷ $M$ samples, $N$ genes, I index set of $n$ regulators
Ensure: $b, L$   ▷ Number of bootstrapping runs and ARNI steps
Initialize $S \in \mathbb{R}^{N \times n}$   ▷ Initialize adjacency matrix of the GRN
Initialize $SM \in \mathbb{R}^{n \times b}$   ▷ Initialize the selection matrix
for $i = 1 \to b$ do   ▷ For each bootstrapping run
    $A^*$ = resample($A$)   ▷ Resampling with replacement
    for $j = 1 \to n$ do   ▷ For each target gene
        $y = A^*_j, X = A^*_{I \setminus j}$
        $SM_{ji}$ = ARNI($y, X, L$)   ▷ Returns selected tx-factors with the ARNI algorithm
    end for
end for
$S$ = area-score($SM, L, b$)   ▷ Get the weight score matrix with the area-score metric
Output: $S$   ▷ Output the score matrix
```

and area-score techniques will be described later in this section.

Feature selection with ARNI

For a given unit i and its corresponding differential equation, ARNI turns to obtain which units j of the network provides direct physical interactions and appears on the right-hand side of the equation, rather than asking for details of the interaction functions among those units in the equation.

In detail, for a dynamic system with N units, ARNI first decomposes unit i's dynamic into interaction terms with other units in the network as [39]:

$$\begin{aligned}\dot{x}_i =& f_i(\Lambda^i x)\\ =& \sum_{j=1}^{N}\Lambda^i_j g^i_j(x_j) + \sum_{j=1}^{N}\sum_{s=1}^{N}\Lambda^i_j\Lambda^i_s g^i_{js}(x_j,x_s)\\ &+\sum_{j=1}^{N}\sum_{s=1}^{N}\sum_{w=1}^{N}\Lambda^i_j\Lambda^i_s\Lambda^i_w g^i_{jsw}(x_j,x_s,x_w)+\ldots+\epsilon_i\end{aligned} \tag{4}$$

where $\dot{x}_i := [\dot{x}_{i,1},\dot{x}_{i,2},\ldots,\dot{x}_{i,M}] \in \mathbb{R}^M$, $f : \mathbb{R}^N \rightarrow \mathbb{R}$ is a smooth function, the diagonal matrices $\Lambda^i \in \{0,1\}^{N\times N}$ and $\Lambda^i_j = 1$ if unit j directly acts on unit i, otherwise $\Lambda^i_j = 0$, $g^i_j : \mathbb{R} \rightarrow \mathbb{R}$, $g^i_{js} : \mathbb{R}^2 \rightarrow \mathbb{R}$, $g^i_{jsw} : \mathbb{R}^3 \rightarrow \mathbb{R}$ and in general $g^i_{j_1 j_2 \ldots j_K} : \mathbb{R}^K \rightarrow \mathbb{R}$ represents the (unknown) K-th order interactions between units j_k for all $k \in \{1,2,\ldots,K\}$ and unit i, the last term ϵ_i represents external noise acting on i.

The functions $g^i_{j_1 j_2 \ldots j_K}$ are not accessible, they can be decomposed into basis functions h, and we can rewrite Eq. (4) as [39]:

$$\begin{aligned}\dot{x}_i =& \sum_{j=1}^{N}\Lambda^i_j\sum_{p=1}^{P_1}c^i_{j,p}h_{j,p}(x_j)\\ &+\sum_{j=1}^{N}\sum_{s=1}^{N}\Lambda^i_j\Lambda^i_s\sum_{p=1}^{P_2}c^i_{js,p}h_{js,p}(x_j,x_s)\\ &+\sum_{j=1}^{N}\sum_{s=1}^{N}\sum_{w=1}^{N}\Lambda^i_j\Lambda^i_s\Lambda^i_w\sum_{p=1}^{P_3}c^i_{jsw,p}h_{jsw,p}(x_j,x_s,x_w)\\ &+\ldots+\epsilon_i\end{aligned} \tag{5}$$

where P_k indicates the number of basis functions employed in the expansion [45], $c^i_{j,p}$, $c^i_{js,p}$, $c^i_{jsw,p}$ are the unknown coefficients. Appropriate basis functions h are favored to form a relevant function space. For instance, the class of pairwise basis functions $g^i_{ij}(x_i,x_j)$ can be in the form of $h^i_{ij,p}(x_i,x_j) = (x_j - x_i)^p$ or $h^i_{ij,p}(x_i,x_j) = x_i^{p_1}x_j^{p_2}$, etc..

Note that the framework is intended to reveal units direct interactions in dynamic systems with time series data especially. For GRN inference problem especially from non-time series data, a minor modification can be applied to Eq. (5). More specially, after replacing the left hand side time varying term $\dot{x}_i$ of the Eq. (5) with a non-time varying term x_i, which is still a vector, and not accounting for self interaction meanwhile, we can have a modified equation defined as:

$$\begin{aligned}x_i =& \sum_{j=1}^{N}\Lambda^i_j\sum_{p=1}^{P_1}c^i_{j,p}h_{j,p}(x_j)\\ &+\sum_{j=1}^{N}\sum_{s=1}^{N}\sum_{w=1}^{N}\Lambda^i_j\Lambda^i_s\Lambda^i_w\sum_{p=1}^{P_3}c^i_{jsw,p}h_{jsw,p}(x_j,x_s,x_w)\\ &+\ldots+\epsilon_i\end{aligned} \tag{6}$$

The transformation from Eq. (5) to Eq. (6) is reasonable, and in this manner Eq. (6) is then the detail implementation of Eq. (3). The reconstruction problem then becomes identifying the non-zero interaction terms in Eq. (6). The vector of coefficients $c^i_{j,p}$, $c^i_{js,p}$, $c^i_{jsw,p}$ are unknown, hindering the retrieval of Λ^i. It is sufficient to impose a structure of blocks of zero and non-zero coefficients in Eq. (6), representing absent and existing interactions, respectively. These structured solutions are composed by blocks c^i_s of non-zero entries (representing the non-zero interactions acting on unit i) distributed along c^i. The Algorithm for Revealing Network Interactions (ARNI) is proposed to solve this mathematical regression problem with grouped variables, which is a greedy approach based on the Block Orthogonal Least Squares (BOLS) algorithm [46]. ARNI can be viewed as a proper feature selection method in essence, the same as the well-known sparse group lasso [47]. Details of the algorithm are explained in the supplementary note of [39].

Bootstrapping

The D3GRN algorithm uses bootstrapping towards to obtain a more reliable selection of the regulators of a target gene. Bootstrapping [48] generates multiple sets of samples from the observed samples by resampling, and then computes the parameter of interest for each resampled set. Finally, an estimate of the parameter in question is obtained by averaging over all of the resampled sets. In resampling, samples are randomly selected (uniformly at random, with replacement) from the observed samples. The bootstrapping technique is frequently applied to get stable results in the case of underdetermined problems [49]. In the current D3GRN implementation, the bootstrapping runs b = 200 times. In each bootstrapping runs, y and X are chosen uniformly at random from resampling with replacement from the given gene expression data. Subsequently, the ARNI algorithm is utilized to select the regulators for each of these bootstrapping runs. Finally, the results of all bootstrapping runs are aggregated using an area-based scoring [34] technique. Note that the D3GRN algorithm applies bootstrapping only to obtain the high-confidence regulators for each target gene, and it does not aggregate over many bootstrapping networks such as that in [34].

Area-Based scoring

The area-based scoring method [37] is to assign a score to each candidate regulator with the frequency of its selection over specified bootstrapping runs. In each bootstrapping run, the ordered list of the regulators of a target gene provided by the ARNI is mathematically independent. This scoring method aims to exploit the overall ordering information about the selection of the regulators. This is achieved via the area based scoring method as follows.

Let ϕ_{ijl} be cumulative selection frequency of j-th regulator in the l-th ARNI step, $l = \{1, \ldots, L\}$ and apparently $\phi_{ijl} \in [0, 1]$. The average is taken over all bootstrapping runs, and the score S_{ij} for regulator i of gene i in total L steps is defined as:

$$S_{ij} = \frac{1}{L}\sum_{l=1}^{L}\phi_{ijl} \tag{7}$$

For example, given the values $\phi_{ij1} = 0.3$, $\phi_{ij2} = 0.5$, and $L = 5$, the j-th regulator was selected 30 percent of the time in the first ARNI step and 20 percent of the time in the second ARNI step in the 5 steps. Then the cumulative selection frequency ϕ_{ij2} is 50 percent. The score S_{ij} has a natural interpretation of an area under the cumulative selection frequency curve normalized by the total area L. Clearly, this score not only takes into account the overall selection frequency of a transcription factor but also rewards the selection in the earlier ARNI steps. This method is less sensitive to the number of ARNI steps than simple ranking based on overall selection frequency ϕ_{ij}.

Results

Input data

GRNs inference has been quite an active area of research during the past decade. Consequently, a community based consortium called “Dialogue for Reverse Engineering Assessments and Methods” (DREAM) [43] is founded. The DREAM consortium holds international reverse engineering challenges, providing standardized common input datasets and performance evaluation metrics to compare different approaches. The DREAM datasets have become a standard benchmark in the GRN inference community and are frequently used to evaluate the performance of new algorithms.

Table 2 Detail of the datasets

Network	#Genes	#Regulators	#Samples	#Verified interactions
DREAM4 Network 1	100	100	100	176
DREAM4 Network 2	100	100	100	249
DREAM4 Network 3	100	100	100	195
DREAM4 Network 4	100	100	100	211
DREAM4 Network 5	100	100	100	193
DREAM5 Network 1	1643	195	805	4012

In our experiments, we use six in-silico datasets in total from both DREAM4 and DREAM5 challenges [50]. The details of the datasets are summarized in Table 2, in which columns stand for the number of genes, the number of regulators (regulatory genes), the number of samples (experiments) and the number of verified interactions respectively. If a dataset is arranged with a matrix, samples mean rows and genes mean columns. We employ five multifactorial datasets from DREAM4 challenge, with each containing 100 genes and 100 samples. The samples in these five datasets are generated from the original data by slightly perturbing all gene expression values at the same time, with the aid of the open-source GeneNetWeaver software [51]. Hence, each sample in the five datasets stands for a multifactorial perturbation experiment. Regulators can be viewed as themselves as lack of regulators provided in these small networks. We also employ one DREAM5 dataset Network 1, which is also a simulated network generated by GeneNetWeaver. The topology of the in-silico network is based on known GRNs of model organisms. Differently from that in DREAM4, The transcription factors (TFs) in DREAM5 datasets are provided as regulators which is a subset among all the genes.

Performance evaluation metrics

To evaluate the performance of the GRN inference algorithms, we use the area under the Precision-Recall curve (AUPR) as an evaluation metric. Together with AUPR, the area under the Receiver Operating Characteristic curve (AUROC) is also widely adopted for performance evaluation. In general, higher AUROC and AUPR value indicate more accurate GRN predictions. It should be noticed that, in sparse biological networks, the number of non-existing edges (negatives) outweighs the number of existing edges (positives) significantly, AUPR is more informative than AUROC [52].

We first compute the numbers of true positive (TP), true negative (TN), false positive (FP) and false negative (FN) edges by comparing the regulatory edges in the gold standard network with the top q edges from the ranked list output of D3GRN. The Precision-Recall curve is constructed by plotting the precision $\frac{\mathrm{TP}}{\mathrm{TP+FP}}$ versus the recall $\frac{\mathrm{TP}}{\mathrm{TP+FN}}$ for increasing q, $q = 1, 2, \ldots, N \times (N-1)$, where N is the number of genes. AUPR is then obtained by calculating the area under the curve.

Performance of D3GRN

The type of basis function, order K and the number of basis functions P_k in Eq. (5) play a critical role in the model decomposition in ARNI. For a large class of dynamic systems, the polynomial nonlinearities are sufficient [53]. As a reference, for gene regulatory network reconstruction in our study, the polynomial basis functions are also employed in the form of $h_{j,p}(x_j) = x_j^p$, and the number of basis functions is denoted as:

$$P_k = \begin{cases} 5, k = 1 \\ 0, k > 1 \end{cases} \tag{8}$$

which means implicitly that we do not consider 2-th and above order interactions for a target gene. In fact, bLARS [37] only considers one order interaction. We also follow this way of simplification in this study. In other words, gene regulation of other genes to a target gene is a mixture of basis polynomial nonlinearities functions.

There are two parameters in D3GRN, including the number of bootstrapping runs b and the number of ARNI steps L. Figure 1 shows the effect of these two parameters by varying the number of ARNI steps and the number of bootstrapping runs from the DREAM5 Network 1. Generally, a larger number of bootstrapping runs b can improve the score with a sacrifice of running time. However, the performance of D3GRN is quite robust to the number of bootstrapping runs provided it is larger than a certain threshold, typically 200 runs. For the ARNI steps L, one's intuition indicates that the performance would be optimal if L is close to the true average number of regulators in the network, which can be obtained with $\frac{2\times\#\text{Verified interactions}}{\#\text{Genes}}$.

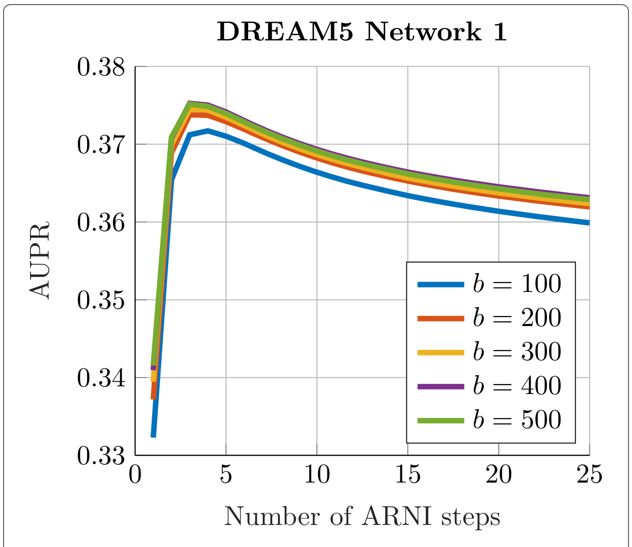

Fig. 1 AUPR by varying ARNI steps L and the bootstrapping number b in DREAM5 Network 1

We have conducted two comparison experiments on DREAM4 and DREAM5 networks, to evaluate our proposed method D3GRN. NIMEFI was implemented with R, while GENIE3, TIGRESS, and PLSNET were implemented in Matlab. The codes are downloaded from the URLs provided in the corresponding papers, and we use the default values of the parameters in each method for performance comparison. Our proposed method D3GRN is also implemented in Matlab, which is available at https://github.com/chenxofhit/D3GRN.

Table 3 lists the results of D3GRN compared with other GRN inference methods on the five DREAM4 networks. In the table, the performance of D3GRN is determined with the bootstrapping number $b = 200$, the number of ARNI steps $L = 2$. D3GRN achieves the best AUPR value except on DREAM4 Network 2.

Table 4 summarizes the results of D3GRN compared with other GRN inference methods on the DREAM5 dataset. The result of D3GRN is obtained with parameters setting as the bootstrapping number $b = 200$, the number of ARNI steps $L = 5$ for Network 1. D3GRN achieves the highest AUPR value on Network 1.

Discussion and conclusion

It is reasonable to assume that interaction structure is sparse in GRN inference. Specially, under the case of "small n large p", i.e. the small number of available samples and the large number of genes, sparsity constraints are widely considered in machine learning. In GRN, the sparsity assumption means that every gene has only a small number of regulators, which seems quite reasonable. The proposed D3GRN method also follows the same assumption. We evaluate our method on the DREAM4 and DREAM5 datasets. We hold the view that gene regulation of other genes to a target gene is a mixture of basis polynomial nonlinearities functions, which is also confirmed by the performance of our method in some extent. Theoretical or experimental analysis of this adoption is left for future work.

Table 3 Performance comparisons of different GRN inference methods on the DREAM4 networks in terms of AUPR

Method	Network 1	Network 2	Network 3	Network 4	Network 5
GENIE3	0.161	0.154	0.234	0.211	0.200
TIGRESS	0.158	0.161	0.233	0.225	0.233
NIMEFI	0.157	0.157	0.248	0.225	0.241
PLSNET	0.118	0.290	0.202	0.228	0.206
D3GRN	0.175	0.136	0.253	0.255	0.247

Table 4 Performance comparisons of different GRN inference methods on the DREAM5 Network 1 in terms of AUPR

Network	GENIE3	TIGRESS	NIMEFI	PLSNET	D3GRN
Network 1	0.291	0.302	0.298	0.270	0.373

Another important issue is about the computational complexity of D3GRN. Speaking objectively, ARNI is suitable for small physical dynamic network recovery from the unit level. The Moore-Penrose pseudo-inverse operation of the BOLS algorithm adopted by ARNI is time consuming for large biological networks. The bootstrapping strategy in D3GRN makes it worse when dealing with large scale GRNs inference. Concerning the improvement space of ARNI, "for" loops in the bootstrapping strategy in D3GRN are completely parallelizable and can be carried out simultaneously on multiple cores and even on distributed machines in a cluster. It also deserves a try with other methods such as BOMP [46] to replace the BOLS algorithm, which is also left for future work.

The variability of the performance of the current state-of-the-art algorithms indicates that there is no algorithm that performs equally well on all datasets. However, all of these algorithms can be applied to provide inputs to a meta-algorithm that takes advantage of "the wisdom of crowds" to create a consensus and reliable community network [54, 55]. Also, the decreasing performance of all the algorithms from small networks to large networks perhaps reflects the increasing complexity of the underlying regulatory networks with varying scales. Our method advances the current state of the art, but there is still a long way to go before the issue could be treated as completely solved.

Constructing GRNs from gene expression data is an important task that can potentially contribute to our understanding of the basic mechanism such as diseases and cancers in system biology. Recent data driven dynamic networks construction methods have opened new possibilities for us to infer GRNs. In this study, we propose a data driven dynamic network construction method to infer gene regulatory networks, which transforms the regulatory relationship of each target gene into a functional decomposition problem and solves it by using the Algorithm for Revealing Network Interactions (ARNI). However, traditional data driven dynamic network recovery methods such as SINDy and ARNI do not have the ability of constructing a network. To address this limitation, we use bootstrapping and area based scoring strategy to obtain a final GRN. On DREAM4 and DREAM5 benchmark datasets, D3GRN performs competitively in terms of AUPR.

Abbreviations

ARNI: The Algorithm for Revealing Network Interactions; AUPR: The area under the Precision-Recall curve; D3GRN: A data driven dynamic network construction method to infer gene regulatory network; GRNs: Gene regulatory networks

Acknowledgements

The authors thank Dr. Jose Casadiego, the author of ARNI algorithm, for his constructive and fruitful discussions. The comments of the anonymous referees, as well as the Associated Editor, are gratefully acknowledged.

About this supplement

This article has been published as part of BMC Genomics Volume 20 Supplement 13, 2019: Proceedings of the 2018 International Conference on Intelligent Computing (ICIC 2018) and Intelligent Computing and Biomedical Informatics (ICBI) 2018 conference: genomics. The full contents of the supplement are available online at https://bmcgenomics.biomedcentral.com/articles/supplements/volume-20-supplement-13.

Authors' contributions

CX conceived the study, designed the experiments and wrote the manuscript; LM supervised the research and extensively edited the manuscript; ZRQ proposed constructive suggestion about the detail of the experiments; WFX and WJX revised the manuscript. All authors read and approved the final manuscript.

Funding

This research is supported in part by the National Natural Science Foundation of China under Grants (No. 61622213, No. 61732009 and No. 61772552), the 111 Project (No. B18059) and the Hunan Provincial Science and Technology Program (2018WK4001).

Availability of data and materials

A Matlab implementation of D3GRN is available at https://github.com/chenxofhit/D3GRN. The datasets of DREAM4 are available on http://www.synapse.org/#\protect\kern-.1667em\relaxSynapse:syn3049712/files/, and the datasets of DREAM5 are available on http://www.synapse.org/#\protect\kern-.1667em\relaxSynapse:syn2787209/files/.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1] School of Computer Science and Engineering, Central South University, Changsha, China. [2] Department of Mechanical Engineering and Division of Biomedical Engineering, University of Saskatchewan, SK S7N 5A9 Saskatoon, Canada.

Published: 27 December 2019

References

1. Lee TI, Rinaldi NJ, Robert F, Odom DT, Bar-Joseph Z, Gerber GK, et al. Transcriptional regulatory networks in Saccharomyces cerevisiae. Science. 2002;298(5594):799–804.
2. Harbison CT, Gordon DB, Lee TI, Rinaldi NJ, Macisaac KD, Danford TW, et al. Transcriptional regulatory code of a eukaryotic genome. Nature. 2004;431(7004):99.
3. Matsumoto H, Kiryu H, Furusawa C, Ko MS, Ko SB, Gouda N, et al. SCODE: an efficient regulatory network inference algorithm from single-cell RNA-Seq during differentiation. Bioinformatics. 2017;33(15):2314–21.
4. Chen H, Guo J, Mishra SK, Robson P, Niranjan M, Zheng J. Single-cell transcriptional analysis to uncover regulatory circuits driving cell fate decisions in early mouse development. Bioinformatics. 2014;31(7):1060–6.
5. Trapnell C, Cacchiarelli D, Grimsby J, Pokharel P, Li S, Morse M, et al. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. Nat Biotechnol. 2014;32(4):381.

6. Boyle EA, Li YI, Pritchard JK. An expanded view of complex traits: from polygenic to omnigenic. Cell. 2017;169(7):1177–86.
7. Maetschke SR, Madhamshettiwar PB, Davis MJ, Ragan MA. Supervised, semi-supervised and unsupervised inference of gene regulatory networks. Brief Bioinformatics. 2013;15(2):195–211.
8. Karlebach G, Shamir R. Modelling and analysis of gene regulatory networks. Nat Rev Mol Cell Biol. 2008;9(10):770.
9. Le Novere N. Quantitative and logic modelling of molecular and gene networks. Nat Rev Genet. 2015;16(3):146.
10. Huynh-Thu VA, Sanguinetti G. Gene regulatory network inference: an introductory survey. arXiv preprint arXiv:180104087. 2018.
11. Stuart JM, Segal E, Koller D, Kim SK. A gene-coexpression network for global discovery of conserved genetic modules. Science. 2003;302(5643): 249–55.
12. Steuer R, Kurths J, Daub CO, Weise J, Selbig J. The mutual information: detecting and evaluating dependencies between variables. Bioinformatics. 2002;18(suppl_2):S231–40.
13. Uda S, Saito TH, Kudo T, Kokaji T, Tsuchiya T, Kubota H, et al. Robustness and compensation of information transmission of signaling pathways. Science. 2013;341(6145):558–61.
14. Mc Mahon SS, Lenive O, Filippi S, Stumpf MP. Information processing by simple molecular motifs and susceptibility to noise. J R Soc Interface. 2015;12(110):20150597.
15. Basso K, Margolin AA, Stolovitzky G, Klein U, Dalla-Favera R, Califano A. Reverse engineering of regulatory networks in human B cells. Nat Genet. 2005;37(4):382.
16. Zhang X, Zhao XM, He K, Lu L, Cao Y, Liu J, et al. Inferring gene regulatory networks from gene expression data by path consistency algorithm based on conditional mutual information. Bioinformatics. 2011;28(1):98–104.
17. Chen X, Li M, Zheng R, Zhao S, Wu F, Li Y, et al. A novel method of gene regulatory network structure inference from gene knock-out expression data. Tsinghua Sci Technol. 2019;24(4):446–55.
18. Zhao J, Zhou Y, Zhang X, Chen L. Part mutual information for quantifying direct associations in networks. Proc Nat Acad Sci. 2016;113(18):5130–5.
19. Chan TE, Stumpf MP, Babtie AC. Gene regulatory network inference from single-cell data using multivariate information measures. Cell Syst. 2017;5(3):251–67.
20. Kauffman S. Homeostasis and differentiation in random genetic control networks. Nature. 1969;224(5215):177.
21. Shmulevich I, Dougherty ER, Kim S, Zhang W. Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. Bioinformatics. 2002;18(2):261–74.
22. Liang J, Han J. Stochastic Boolean networks: an efficient approach to modeling gene regulatory networks. BMC Syst Biol. 2012;6(1):113.
23. Friedman N, Linial M, Nachman I, Pe'er D. Using Bayesian networks to analyze expression data. J Comput Biol. 2000;7(3-4):601–20.
24. Friedman N, Koller D. Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks. Mach Learn. 2003;50(1-2):95–125.
25. Grzegorczyk M, Husmeier D. Improvements in the reconstruction of time-varying gene regulatory networks: dynamic programming and regularization by information sharing among genes. Bioinformatics. 2010;27(5):693–99.
26. Hill SM, Lu Y, Molina J, Heiser LM, Spellman PT, Speed TP, et al. Bayesian inference of signaling network topology in a cancer cell line. Bioinformatics. 2012;28(21):2804–10.
27. Chen T, He HL, Church GM. Modeling gene expression with differential equations. In: Biocomputing'99. World Scientific; 1999. p. 29–40.
28. Bonneau R, Reiss DJ, Shannon P, Facciotti M, Hood L, Baliga NS, et al. The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. Genome Biol. 2006;7(5):R36.
29. Kikuchi S, Tominaga D, Arita M, Takahashi K, Tomita M. Dynamic modeling of genetic networks using genetic algorithm and S-system. Bioinformatics. 2003;19:643–50.
30. Wang H, Qian L, Dougherty E. Inference of gene regulatory networks using S-system: a unified approach. IET Syst Biol. 2010;4(2):145–56.
31. Liu LZ, Wu FX, Zhang WJ. Inference of biological S-system using the separable estimation method and the genetic algorithm. IEEE/ACM Trans Comput Biol Bioinformatics. 2012;9(4):955–65.
32. Huynh-Thu VA, Irrthum A, Wehenkel L, Geurts P. Inferring Regulatory Networks from Expression Data Using Tree-Based Methods. PLoS ONE. 2010;5(9):e12776.
33. Marbach D, Prill RJ, Schaffter T, Mattiussi C, Floreano D, Stolovitzky G. Revealing strengths and weaknesses of methods for gene network inference. Proc Nat Acad Sci. 2010;107(14):6286–91.
34. Haury AC, Mordelet F, Vera-Licona P, Vert JP. TIGRESS: Trustful Inference of Gene REgulation using Stability Selection. BMC Syst Biol. 2012;6(1):145.
35. Ruyssinck J, Geurts P, Dhaene T, Demeester P, Saeys Y, et al. Nimefi: gene regulatory network inference using multiple ensemble feature importance algorithms. PLoS One. 2014;9(3):e92709.
36. Zou H, Hastie T. Regularization and variable selection via the elastic net. J R Stat Soc B (Stat Methodol). 2005;67(2):301–20.
37. Singh N, Vidyasagar M. bLARS: an algorithm to infer gene regulatory networks. IEEE/ACM transactions on computational biology and. Bioinformatics. 2016;13(2):301–14.
38. Brunton SL, Proctor JL, Kutz JN. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc Nat Acad Sci. 2016;113(15):3932–7.
39. Casadiego J, Nitzan M, Hallerberg S, Timme M. Model-free inference of direct network interactions from nonlinear collective dynamics. Nat Commun. 2017;8(1):2192.
40. Sławek J, Arodź T. ENNET: inferring large gene regulatory networks from expression data using gradient boosting. BMC Syst Biol. 2013;7(1):106.
41. Guo S, Jiang Q, Chen L, Guo D. Gene regulatory network inference using PLS-based methods. BMC Bioinformatics. 2016;17(1):545.
42. Zheng R, Li M, Chen X, Zhao S, Wu F, Pan Y, et al. An ensemble method to reconstruct gene regulatory networks based on multivariate adaptive regression splines. IEEE/ACM Trans comput Biol Bioinformatics. 2019.
43. Stolovitzky G, Monroe D, Califano A. Dialogue on Reverse-Engineering Assessment and Methods. Ann N Y Acad Sci. 2007;1115(1):1–22.
44. Nasrabadi NM. Pattern recognition and machine learning. J Electron Imaging. 2007;16(4):049901.
45. Friedman J, Hastie T, Tibshirani R. The elements of statistical learning. vol. 1, No.10. New York: Springer series in statistics; 2001.
46. Majumdar A, Ward RK. Fast group sparse classification. Can J Electr Comput Eng. 2009;34(4):136–44.
47. Friedman J, Hastie T, Tibshirani R. A note on the group lasso and a sparse group lasso. arXiv preprint arXiv:10010736.
48. Johnson RW. An introduction to the bootstrap. Teach Stat. 2001;23(2): 49–54.
49. Wang S, Nan B, Rosset S, Zhu J. Random lasso. Ann Appl Stat. 2011;5(1): 468.
50. Marbach D, Costello JC, Küffner R, Vega NM, Prill RJ, Camacho DM, et al. Wisdom of crowds for robust gene network inference. Nat Methods. 2012;9(8):796.
51. Marbach D, Schaffter T, Mattiussi C, Floreano D. Generating realistic in silico gene networks for performance assessment of reverse engineering methods. J Comput Biol. 2009;16(2):229–39.
52. Saito T, Rehmsmeier M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PloS One. 2015;10(3):e0118432.
53. Mangan NM, Brunton SL, Proctor JL, Kutz JN. Inferring biological networks by sparse identification of nonlinear dynamics. IEEE Trans Mol Biol Multi-Scale Commun. 2016;2(1):52–63.
54. Vera-Licona P, Marbach D, Irrthum A, Prill RJ, Haury AC, de la Fuente A, et al. Wisdom of crowds for robust gene network inference. Nat Methods. 2012;9(8):796–804.
55. Zheng CH, Huang DS, Kong XZ, Zhao XM. Gene expression data classification using consensus independent component analysis. Genomics Proteomics Bioinformatics. 2008;6(2):74–82.

Publisher's Note