Journal of Biomedical Informatics 65 (2017) 120–131

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Variable neighborhood search for reverse engineering of gene regulatory networks

Charles Nicholson [a,*], Leslie Goodwin [a], Corey Clark [b]

[a] School of Industrial and Systems Engineering, University of Oklahoma, Norman, OK, United States
[b] Guidhall, Southern Methodist University, Dallas, TX, United States

ARTICLE INFO

Article history:
Received 4 December 2015
Revised 16 November 2016
Accepted 27 November 2016
Available online 2 December 2016

Keywords:
Gene regulatory networks
Reverse engineering
Variable neighborhood search

ABSTRACT

A new search heuristic, Divided Neighborhood Exploration Search, designed to be used with inference algorithms such as Bayesian networks to improve on the reverse engineering of gene regulatory networks is presented. The approach systematically moves through the search space to find topologies representative of gene regulatory networks that are more likely to explain microarray data. In empirical testing it is demonstrated that the novel method is superior to the widely employed greedy search techniques in both the quality of the inferred networks and computational time.

1. Background

A gene regulatory network (GRN) is a collection of genes, regulators, and regulatory connections that govern expression levels [1]. Analysis of GRNs has become essential for better understanding cellular systems because it provides insight into which genes control the activation of others [2,3]. The network topology has various interpretations in literature: the nodes in the GRN may represent genes or their protein products, the undirected edges between nodes may indicate genes are co-regulated, share common functionality, location or process, or directly bind one another; and directed edges may imply a step in a metabolic pathway, signal transduction cascade, stage of development, or a causal relationship [4]. These networks create the blackprint of the cellular system structure and provide design details of the cell.

Research in computational systems biology revolves around inferring or reverse engineering GRNs based on gene expression levels [5]. A basic assumption within the field is that the observed data, which are the changes in mRNA expression profiles, can explain transcriptional regulation. By inferring the underlying gene regulatory network from these large-scale experiments, ultimately the molecular role can be understood. The expression levels are the output of specific gene regulatory networks and therefore many algorithms have been studied to reverse engineer the GRNs most likely to produce observed expression data. Numerous issues arise from modeling GRNs from experimental data and therefore no one modeling technique outperforms all others. There are a vast number of genes and potential relationships; the experimentation to measure expression levels often result in noisy data; and there may be unobserved factors affecting the activity of genes that are not represented in the experiments conducted [1,6]. Once the networks are modeled, the topologies are scored to determine which are most consistent with the data. However, even the simplest GRNs are complex systems and difficult to infer.

Active research in reverse engineering of GRNs is conducted by testing different mathematical methods on computer generated networks where the true network is known. This allows for both validation and analysis of various inference algorithms. There are a few notable models commonly used for inferring GRNs: boolean networks [7], differential equations and linearization [8], regression methods [9], Gaussian models [10], conditional correlation analysis [11], and static and dynamic Bayesian networks [12,13]. Each provides advantages and disadvantages when inferring topologies [14]. Ultimately, the goal is to reverse engineer networks with confidence that the output of the statistical model is representative of the biological system.

1.1. Bayesian networks

Bayesian network (BN) modeling is an approach that combines probability and graph theory which has been useful in recovering gene regulatory networks from data. They can be used to describe the relationship between variables in gene regulatory networks and are promising because they can capture multiple types of

[*] Corresponding author.
E-mail addresses: cnicholson@ou.edu (C. Nicholson), leslie.goodwin@ou.edu (L. Goodwin), coreyc@smu.edu (C. Clark).

http://dx.doi.org/10.1016/j.jbi.2016.11.010
1532-0464/© 2016 Elsevier Inc. All rights reserved.

relationships [15]. These networks describe the relationship at a qualitative level. At the qualitative level, the graphical model showcases the dependences between various genes, which are encoded in the structure of the directed graph. An example BN is depicted in Fig. 1 in which $\mathbf{X} = (X_1, \ldots, X_5)$ represents the genes and the edges represent the dependencies. Each term $p(X_i|\mathrm{PA}_i)$ is the probability for a variable conditioned on the set of parents PA_i of X_i. Bayesian networks specify the joint distribution over all variables for the conditional distribution of the node given the parental relationship:

$$p(X_1, \ldots, X_n) = \prod_{i=1}^{n} p(X_i|PA_i).$$

Numerous experiments have been conducted on *in silico* data to compare Bayesian networks to other inference models. Margolin et al. [16] developed ARACNE (algorithm for the reconstruction of accurate cellular networks) as another algorithms for inferring GRNs. Their study compares ARACNE to BNs because BNs are so widely used in reverse engineering and as such, the authors claim they provide an ideal benchmark technique. BNs are among the most effective models because of their ability to account for the stochastic nature of gene expression profiles and the easy integration of prior knowledge [14,17].

BNs are directed acyclic graphs and therefore the topology produced by the predicted model will include directed edges. This allows for modeling gene expression levels which depend on the regulators (parents) in the network. Accurate directed predictions are more difficult than undirected predictions. Some algorithms (e.g. ARACNE) only produce undirected results since the undirected topologies still provide useful insight into the underlying structure.

Given observed expression data D, a Bayesian network approach enables a quantitative assessment regarding the likelihood that directed graph G produces such data. The general Bayesian scoring metric from [1] is the posterior probability of graph G given D:

$$\begin{aligned} S(G:D) &= \log P(G|D) = \log \frac{p(D|G)p(G)}{p(D)} \\ &= \log p(D|G) + \log p(G) + \text{constant}. \end{aligned} \tag{1}$$

The goal is to maximize the Bayesian score in Eq. (1). This score provides the ability to evaluate the quality of candidate graphs when searching for the network topology. In particular, we employ the Bayesian Dirichlet Equivalence (BDe) score [18,19] to help learn the BN and evaluate candidate GRN. This score incorporates a likelihood equivalence assumption and also allows for the incorporation of prior knowledge [19]. If relationships between nodes are already known, this information can be incorporated into the model. The metric penalizes any graph not containing an edge provided in the prior network. Another advantage of the score is the penalization of overly complex structures and the preference of simpler models of equally good networks.

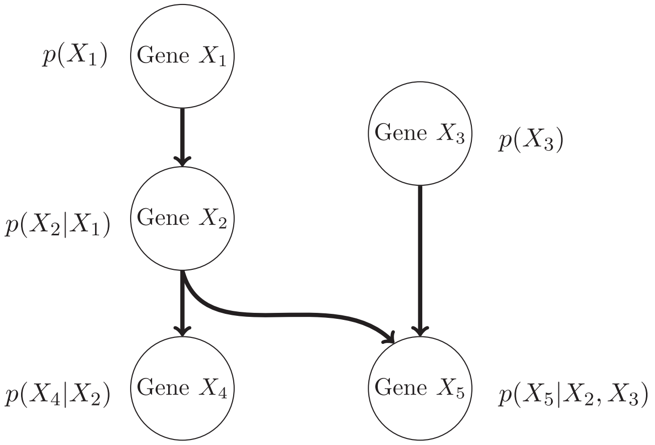

Fig. 1. Bayesian network example.

1.2. Search heuristics

Finding the network topology that maximizes the likelihood of expressing the observed data is NP-hard [20,21]. Since the search space is large and no efficient exact algorithms are known for this problem, heuristic search is commonly used. The goal of heuristic search is to find a near optimal solution quickly and efficiently.

One commonly used search heuristic is the greedy technique *hill climbing* [6,22,23]. Hill climbing is similar to gradient ascent except that no derivatives are necessary. Instead, this iterative approach evaluates solutions that are "near" the current solution and adopts a new solution if a better one is found in the local search space. Compared with other techniques, this greedy search is fast, computationally simple, and requires few tuning parameters. Hill climbing, however, is myopic and prone to premature convergence to poor local optima. *Random restarts* are incorporated to mitigate this issue and expand the search region by performing hundreds or thousands of hill climbing procedures from randomly generated initial locations in the search space [24]. Yu et al. [15] found hill climbing with random restarts superior to simulated annealing and genetic algorithms. Other local search methods have been applied to learning Bayesian networks outside of the scope of gene regulatory networks: genetic algorithms [25], tabu search [26], ant colony optimization [27], dynamic programming with Markov Chain Monte Carlo techniques [28,29] and swarm optimization [30].

While Bayesian networks continue to be widely studied in application of gene regulatory network inference, research on the search heuristics paired with GRN inference is relatively limited. To date no search algorithms which have been paired with GRN inference have been able to compete with both the speed and solution quality of hill climbing with random restarts. It is the focus of this study to introduce a search heuristic that outperforms this greedy approach without compromising computation time. In this investigation we propose the *Divided Neighborhood Exploration Search* (DNES) heuristic to be paired with the Bayesian network modeling framework and evaluate its performance in producing high quality GRN's.

2. Methods

2.1. In silico data and inference

To accurately evaluate an inference method, the true network must be known. As such, *in silico* data must be used. In particular, a directed acyclic graph, $G = (V, E)$ is constructed where V is the set of nodes, and E is the set of directed edges (i,j), with $i,j \in V$. The constructed topology can then be used to generate data that simulates gene expression data using ordinary differential equations that relate the changes in gene transcript concentration to each gene and to external perturbations. An inference method is used to reverse engineer the original network from the data. In the present study, we compare the implementation of Bayesian networks with a known greedy technique versus the novel DNES algorithm. Since the true network G is known, the quality of the engineered network G' is assessed based on agreement between the topologies of G and G'. Fig. 2 depicts the high-level process.