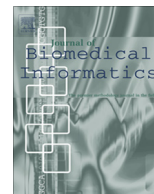




Contents lists available at ScienceDirect

Journal of Biomedical Informatics

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

SAGA: A hybrid search algorithm for Bayesian Network structure learning of transcriptional regulatory networks

Emmanuel S. Adabor^{a,*}, George K. Acquaaah-Mensah^b, Francis T. Oduro^a^a Department of Mathematics, Kwame Nkrumah University of Science and Technology, PMB, Kumasi, Ghana^b Pharmaceutical Sciences Department, Massachusetts College of Pharmacy and Health Sciences (MCPHS University), 19 Foster Street, Worcester, MA, USA

ARTICLE INFO

Article history:

Received 31 January 2014

Accepted 22 August 2014

Available online xxxx

Keywords:

Bayesian Network

Inference

Search algorithms

Transcriptional regulatory network

Microarray dataset

ABSTRACT

Bayesian Networks have been used for the inference of transcriptional regulatory relationships among genes, and are valuable for obtaining biological insights. However, finding optimal Bayesian Network (BN) is NP-hard. Thus, heuristic approaches have sought to effectively solve this problem. In this work, we develop a hybrid search method combining Simulated Annealing with a Greedy Algorithm (SAGA). SAGA explores most of the search space by undergoing a two-phase search: first with a Simulated Annealing search and then with a Greedy search. Three sets of background-corrected and normalized microarray datasets were used to test the algorithm. BN structure learning was also conducted using the datasets, and other established search methods as implemented in BANJO (Bayesian Network Inference with Java Objects). The Bayesian Dirichlet Equivalence (BDe) metric was used to score the networks produced with SAGA. SAGA predicted transcriptional regulatory relationships among genes in networks that evaluated to higher BDe scores with high sensitivities and specificities. Thus, the proposed method competes well with existing search algorithms for Bayesian Network structure learning of transcriptional regulatory networks.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

Bayesian Networks (BNs) are useful for reverse-engineering gene networks from microarray data. BNs are probabilistic in nature. BNs are, therefore, becoming increasingly useful for the inference of cellular networks, modeling signaling pathways and analyzing genetic data [1–3]. Unlike other algorithms for inferring relationships among genes such as the ordinary differential equation (ODE)-based methods which generally involve time series data and low dimensional datasets, BNs (including dynamic BNs) can effectively infer relationships from both large steady-state expression measurements (and time series measurements) [4–8]. Furthermore, clustering algorithms and information-theoretic-based methods only establish the existence of relationships among variables in the dataset but do not describe the specific directionality of relationships [9]. On the other hand, BNs infer causal relationships (when the causal Markov assumption holds) making them more desirable than many inference algorithms.

However, structure learning of BNs from data is an NP-hard problem and so many heuristic methods have been proposed to solve search problems [10,11]. Most of the proposed methods are

local search methods. The number of possible direct acyclic graphs (DAGs) grows exponentially for increasing numbers of nodes [12]. For this reason, we review below a number of search approaches that have been proposed to search for BN structures among DAGs.

1.1. Some current search algorithms

Since finding the optimal BN is NP-hard, most researches have involved reducing the search space of DAGs for algorithms. In the *Sparse Candidate Algorithm*, the super-exponential search space of DAGs is restricted by limiting the parents of each variable to a smaller set using measures of dependencies between variables that are based on a refined Mutual Information measurement [13]. Though this method saves computational time, it does not guarantee an exhaustive search because of the restrictions in the search space. In another approach, the *k-greedy equivalence search algorithm*, different local optima are explored in repeated greedy searches. The randomness of the repeated greedy equivalence search in the *k-greedy equivalence search algorithm* enables it to search good local optima [14]. The approach by Gamez and Puerta restricted the set of parent nodes, took advantage of the properties of locally consistent metrics and removed some nodes from the possible parent nodes as the search proceeded [15]. At the start of the search, no parent set to any variable is restricted permitting

* Corresponding author.

E-mail address: healmes@gmail.com (E.S. Adabor).

all neighbors to be explored. However, some of the variables are removed from the candidate parents based on the metric leading to a restriction on the candidates to be explored. This method requires prior knowledge of the parent sets among the variables. Such prior knowledge is rarely available for prediction. On the other hand, the *A* search algorithm* learns the optimal Bayesian Network by exploring only the most promising parts of the search space [16]. At each step of this search, the node with the smallest cost from the priority queue (OPEN list) is selected to expand its successor nodes before it is placed in a CLOSED list. Although this has improved search time and efficiency, it does not guarantee an exhaustive search as it explores only part of the search space.

Furthermore, heuristic approaches such as the hill-climbing method of the greedy search methods are among the most widely used search strategies for BN structure learning. However, a greedy search is prone to convergence on a local optimal solution which might not necessarily be sufficiently close to the global optimum. Nevertheless, the Greedy search produces fairly good results when the search is repeated several times; starting the search each time with a different randomly generated initial solution [17]. Further, a hybrid approach that combines the greedy search with a reducing search space exists [18]. This involves progressively restricting candidate solutions by identifying a set of forbidden parents to each node, while performing an iterated hill-climbing search [18]. The authors add variables to forbidden nodes if the conditional independence between variables and their parent sets cannot be ignored because of differences in scores as in Gamez et al. [18]. Such an approach reduces the search space for subsequent iterations and hence minimizes the time of search while producing high quality networks.

A version of Simulated Annealing (SA) has also been used to search for good BN structures [19,20]. In this method, the search starts at an initial temperature T_0 , perturbs the existing structure and evaluates it. The new structure is accepted if it improves the network score. However, if the new structure decreases the network score, it is accepted based on a functional value depending on the temperature of the system [19]. The initial temperature is lowered gradually until a putative temperature is reached to decide the termination of the search. Though SA performs well for large datasets, its performance is not improved with increasing time, thus limiting the scores of networks found [20]. Simulated Annealing with re-annealing (SAR) has sought to improve Simulated Annealing by permitting temperature to rise in later parts of the search. This strategy enables the method to escape local optimal solutions. All of these methods are sequential approaches to the search for solutions. However, Jaakkola et al. have also used linear programming techniques to infer Bayesian Network structures [21]. Wang et al. further presented a hybrid method that generates a skeleton BN based on a dependency analysis in a first phase. This is followed by a search that uses a scoring metric combined with the knowledge learned from the first phase [22].

There are also sets of algorithms that mimic natural behavior as they search for solutions without any idea of the nature of expected solutions (such as are expressed as prior networks). These algorithms examine a collection of possible solutions from the search space at each step. One such algorithm, Ant Colony Optimization, as applied to learning BNs, proceeds with an empty graph, and incrementally adds edges. This improves the K2 metric score of the network until a desired stopping criterion is met [23]. Daly and Shen presented a new algorithm, called ACO-E, for Bayesian Network structure learning [24]. This method employed Ant Colony Optimization (ACO) to search the space of possible networks. The authors found the method to outperform the Greedy search and others, while searching in the space of equivalence classes. Larranaga et al. presented a review of evolutionary algorithms

applied to Bayesian Network structure learning [25]. In particular, the most widely used of such methods namely, the Genetic Algorithm (GA), handles populations of candidate solutions to the problem (in contrast to the sequential approaches) across a number of generations in order to obtain better solutions. The algorithm selects a subset of fitter solutions from the population to be the parents to reproduce new offspring solutions via evolution operators. The fittest solutions are then selected as new population for subsequent operations [26,27]. GAs are successful when node ordering assumptions hold, enabling the method to generate acyclic networks. Lee et al. further present a version of GA for BN structure learning where a population of solutions, represented as matrices, are generated from upper and lower triangular matrices together with new genetic operators [28].

The Particle Swarm optimization also considers populations of solutions and selects solutions based on their fitness score at each step. Unlike the GA, it does not require any evolution operators and the K2 metric is used to evaluate such networks [29]. Though these population-based algorithms are able to produce very good results, they do not require known prior relationships among variables at the start of search. Moreover, evaluating every member of a population at each step to determine their fitness is expensive. For these reasons, such algorithms are usually applied to the class of optimization problems where one has no idea about expected optimal solutions [30]. However, for purposes of inferring transcriptional regulatory networks such as is presented in Adabor et al. with BNs (where priors guide the search to find likely relationships among variables), those methods that depend on domain knowledge expressed in prior relationships are definitely preferred [31]. In particular, in the current study, we compare our proposed sequential method, which builds on prior domain knowledge solution, with existing high-performing search methods (which also use prior knowledge represented in non-empty networks to guide the search toward more likely realistic solutions).

At the other end of the spectrum of BN structure learning methods are the constraint-based methods. These use statistical tests of pairs of variables to find conditional independence based on some threshold. The conditional independencies form the basis for the Bayesian Network structure of interest [32,33]. However, the constraint-based methods are sensitive to errors on independence tests. There are hybrid algorithms that aggregate both score-and-search-based and constraint-based approaches of structure learning [34,35]. The Max–Min Hill Climbing algorithm is also a hybrid method that infers a skeleton of Bayesian Network with the constraint-based approach. It then uses a Bayesian scoring Hill-Climbing search to determine the orientations of the edges in the skeleton [36]. This method has been shown to be computationally less expensive than the Sparse candidate algorithm.

1.2. Proposed hybrid approach

From the foregoing, several of the existing sequential approaches for performing BN structure learning basically focus on searching over a reduced search space. This tends to make the algorithms less exhaustive when applied to problems of higher dimensions. In view of this issue, we present in the current paper, a more exhaustive hybrid search method which combines Simulated Annealing with a Greedy Algorithm (SAGA) that thoroughly explores most of the search space by undergoing a two-phase search: first, with a Simulated Annealing search and then with a Greedy search. This approach guarantees near optimal solutions within a fixed time without degrading the quality of the true regulatory network achieved. The random restarts of the greedy algorithm executed in the final phase guarantee a quick convergence to an optimistic best local or near optimal network. The SAGA technique coerces the decomposability of the Bayesian Dirichlet Equivalence (BDe) score function to

evaluate the changes in networks. It also assists in model selection, which enhances the time of the search process [37]. The quick convergence to the local optimal network is further enhanced by the transient and stochastic nature of the search process. Prior domain knowledge is used to guide SAGA so that the results have biological significance and usefulness. Furthermore, the two-phase SAGA search does not restrain the number of nodes for effective structure learning, thus giving equal chances for any given variable to relate to other variables.

2. Methods

2.1. BN structure inference problem

The problem of learning a BN structure is stated as: given a gene expression dataset, D , find a BN model, G , which best describes the dataset where G is in the space of direct acyclic graphs (DAGs). The model graphically describes the relationships among the variables (genes or their products) in a network of regulatory relationships. Given the dataset, the conditional dependencies among the variables are indicated by directed edges between pairs of the variables. Fig. 1 displays an example of a BN.

For any pair of variables, if there is an edge connecting them, then this edge may be reversed or deleted to generate another network. On the other hand, if there is no edge between them, then an edge may be introduced to generate a new network. By these procedures, there are $O(n^2)$ possible changes with n being the number of variables. In a network model representation of a BN, all possible networks (the collection of DAGs) for a given number of variables make up the search space. Inferring Bayesian Networks involves two main parts namely the scoring metric and the search procedure.

2.1.1. Score function

A score function evaluates each BN. The Bayesian scoring metric is described as

$$\text{Score}(G : D) = \log P(G|D) = \log P(D|G) + \log P(G) - \log P(D) \quad (1)$$

where D is an assumed multinomial sample and G is a network, and the marginal likelihood which averages the probability of the data over all possible parameters of G is given by

$$P(D|G) = \int P(D|G, Q)P(Q|G)dQ \quad (2)$$

For the choice of the priors, $P(G)$ and $P(Q|G)$, the Bayesian Dirichlet (BD) metric distinguishes itself by assuming that the priors follow the Dirichlet distribution for each network G [37]. It is given by Eq. (3).

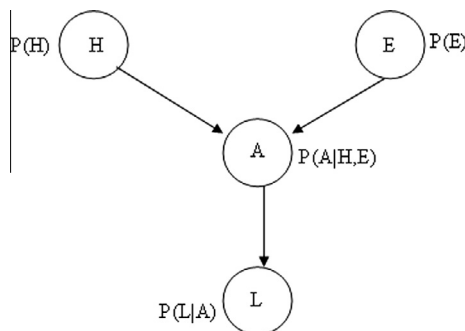


Fig. 1. Example of a Bayesian Network. The network encodes the Joint Probability function $P(H, E, A, L)$ which is given by $P(H, E, A, L) = P(L|A)P(A|H, E)P(H)P(E)$.

$$P(G, D) = P(G) \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{\Gamma(N'_{ij})}{\Gamma(N'_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(N'_{ijk} + N_{ijk})}{\Gamma(N'_{ijk})} \quad (3)$$

where

N'_{ijk} is the Dirichlet distribution orders or exponents for a set of parameters,

$$N'_{ij} = \sum_{k=1}^{r_i} N'_{ijk} \quad (4)$$

$$N_{ij} = \sum_{k=1}^{r_i} N_{ijk} \quad (5)$$

G is BN structure,

D is the dataset,

r_i is the number of states of variable x_i which is equivalent to k ,

q_i is the number of states of Π_i ,

n is number of variables,

N' is equivalent sample size,

N_{ijk} is the number of cases in a database where $x_i = k$ and $\Pi_i = j$

Γ is the Gamma function which is such that $\Gamma(x+1) = x\Gamma(x)$.

Building on the BD metric, Heckerman et al. derived the Bayesian Dirichlet Equivalence (BDe) metric which is based on the likelihood equivalence assumption [38]. The likelihood equivalence assumption states that: “the data does not discriminate between equivalent structures”. The BDe further assumes that the parameters follow the Dirichlet distribution and that the metric has the property of score equivalence. Though it follows Eq. (3), the orders are determined by the equivalent sample size, N' , and

$$N'_{ijk} = N'P(x_i = k, \Pi_i = j|B_{SC}^e, \xi) \quad (6)$$

where Π_i represents the parents of x_i (node i) in the BN structure, B_{SC}^e is the event of complete BN structure and ξ indicates prior knowledge. The probability $P(G)$ which incorporates a penalty factor is given by

$$P(G) = c \prod_{i=1}^n \kappa^{\delta_i} \quad (7)$$

where $0 < \kappa \leq 1$ is the penalty factor, c is a normalization constant and δ_i is the number of nodes in the symmetric difference between the parents of x_i in G and the parents of x_i in the prior network. Thus the metric penalizes G for every edge that differs from edges in the prior network used in the computation. The BDe metric is used in this work since it discriminates between the simple and complex structures in accordance with the Occam's Razor, i.e. preferring the simplest among equally good networks. Heckerman et al. and Cooper and Herskovits discuss details of the construction of the BDe metric [37,38]. The score of each network will enable the algorithm to decide which network to keep for subsequent iterations.

2.1.2. The search method

The proposed search, SAGA, used in the current study is a search over the space comprising of networks that differ from one another by the presence or absence of specific directed edges. The search algorithm makes successive edge changes between variables in the network. It involves randomly generating a new network by making three possible edge changes: addition of edge between any two variables, removal of edge between any two variables, or reversing the directionality of an existing edge between any two variables to achieve a new network in the space of DAGs. Given that each new network is a state and depends only on a unique network in the previous time step or iteration, the search

follows a Markov process. Klebaner provides a description of the properties of Markov processes [39].

More importantly, every member of the DAG space is reachable through a random perturbation of an existing network. The search process does not result in an already examined network since the DAGs differ by edge changes. Thus, the search process is transient, making it impossible for a network which is rejected for poor scores to re-appear. In other words, when a search operation results in a change in relationship between two variables, X_i and X_j for instance, a change from " X_j regulates X_i " to " X_i regulates X_j ", then there cannot be any operation on the network at any subsequent step that will result in " X_j regulates X_i ", except when other concomitant network changes result in a higher-scoring network that includes such an edge. This property allows the search to hasten to convergence as well as perform an efficient search of relationships among variables as it does not visit an already explored relationship, thereby optimizing time and memory usage.

SAGA involves a preliminary exploration of the search space with a search technique guided by a prior network in phase 1 with Simulated Annealing, which is effective at handling large datasets. The prior network is a previously known set of regulatory relationships among the variables usually supplied by domain experts or the literature. This generates a good solution which then serves as input for phase 2 in which there is a greedy search for BN structures. An overview of the SAGA search is as follows:

1. Set initial conditions and prior network (solution).
2. Evaluate current solution with the score function based on the BDe.
3. Generate new solution by randomly performing any of the possible edge changes described.
4. Test conditions to accept new solution. In phase 1, accept the highest scoring new network if it improves score of previous network. Otherwise, compute p (where $p = (\text{changes in the two consecutive network scores})/T_0$, where T_0 is initial temperature). If $p > 1$, accept the new solution, otherwise reject the new solution.
5. Update the initial conditions: $T_0 = \alpha T_0$ where α is a randomly generated number in the interval $(0, 1)$.
6. Repeat steps 3–5 until criterion for terminating phase 1 is met.
7. Start phase 2: Randomly generate a new network from current best network and evaluate it.
8. If the newly generated solution in step 7 has a higher score than the current best network, set the new network as the current best network.
9. Repeat 7–8.
10. Stop.

2.1.3. Termination of search

The algorithm stops when any of the following criteria is met:

- Maximum number of networks explored: Search stops after a fixed number of networks are explored.
- Time elapsed: Search stops after a fixed time.
- Maximum number of restarts: Search stops after a fixed number of restarts of the search process.

In order to avoid a situation where the final phase local search becomes trapped on a local optimal network, the retained best local network structure is randomly perturbed for a different starting network for repeating the search in step 9 of phase 2. This procedure is sometimes called the iterated hill-climbing method [30]. The simplicity of the phase 2, coupled with a good network from phase 1 facilitates a quick convergence to the best network.

3. Results and discussion

3.1. Description of datasets

The data used for the evaluation of SAGA are three microarray datasets obtained respectively from breast cancer patients, human airway smooth muscle cells and mouse whole brain. Similar approaches have been used in the dialogue for Dialogue for Reverse Engineering Assessments and Methods [40–42].

3.1.1. Dataset 1 (breast cancer)

Specifically, a compendium of microarray data was generated from the Gene Expression Omnibus (GEO) record numbers GDS3716, GSE21947 and GDS3139 for the breast cancer dataset. After the raw data (.CEL files format) were downloaded, they were subjected to Robust Multi-array Average processing for normalization and background correction as presented by Irizarry et al. [43]. The subsets of the expression matrices corresponding to variables or probe sets of interest in the dataset were extracted with a Lisp code for use in this work. The dataset on human breast comprised 2453 probe set identifiers (variables) and 89 arrays (see [Supplementary Table S1](#)).

3.1.2. Dataset 2 (human airway smooth muscle)

The human airway smooth muscle (ASM) dataset was obtained from the Gene Expression Omnibus (GEO) repository. Dataset GSE13168 consists of 54 arrays generated from human airway smooth muscle cells from four donors in passage 5–8 [44]. The data represents a number of genetic and pharmacological perturbations in ASMs. After the raw data (.CEL files format) were downloaded, they were subjected to Robust Multi-array Average processing for normalization and background correction as presented by Irizarry et al. [43]. The subsets of the expression matrices corresponding to variables or probe sets of interest in the dataset were extracted with a Lisp code for use in this work. The dataset on ASM comprised 168 probe sets (variables) and 54 arrays (see [Supplementary Table S2](#)).

3.1.3. Dataset 3 (mouse whole brain)

The mouse dataset consists of expression data from 411 mouse microarrays obtained from the Phenogen database and described previously [45,46]. The dataset represents a variety of genetic perturbations of the mouse brain, given the different strains involved. This dataset comprised 2797 variables and 411 arrays (see [Supplementary Table S3](#)).

3.2. Reference networks

A drawback of the components of some current hybrid algorithms for inferring regulatory relationships is that they involve techniques that generally require time series data or data with low dimensional parameter space [8]. Consequently, the choice of Gene Network Inference with Ensemble of trees (GENIE3) [47] to generate reference networks in this work is a good one, especially since the datasets used for the evaluation of SAGA are not time series datasets. GENIE3 models regulatory relationships between n genes as n different regression problems. The expression pattern of each one of the genes (in turn) is predicted from the expression patterns of all the other genes, using tree-based ensemble methods Random Forests or Extra-Trees in each of the regression problems [47]. Two different reference networks were generated with different numbers of genes. Using lists of transcription factors from the human datasets, direct regulatory relationships among the variables were learned with GENIE3 to achieve reference networks to evaluate the algorithm. Such regulatory relationships were

obtained for the human breast cancer datasets (see [Supplementary Table S4](#)) and the ASM datasets (see [Supplementary Table S5](#) where probe set indices is used). In addition, the Context Likelihood Relatedness (CLR) algorithm [48] was used to generate further relationships (see [Supplementary Table S6](#)) from the mouse dataset by computing the dependencies between variables with a refined mutual information measure. These were used as reference networks for the assessment of SAGA.

3.3. Performance indicators

The performance of the SAGA was measured by examining the network inferred from the gene expression datasets by SAGA. Furthermore, the competitiveness of SAGA was assessed with basic and improved existing sequential algorithms such as the Simulated Annealing with re-annealing (SAR) at higher temperature and the Greedy search with random restarts (GR). The basic Greedy search (GS) method initiates with a prior network structure, applies possible changes to the current networks and the resulting network with the highest score is retained as the current network. The search is terminated upon meeting desired stopping criteria. The advanced Greedy search with random restarts (GR) ensures that the search is performed repeatedly each time with different randomly generated initial solution. This overcomes being trapped on a misleading local optimal solution. This approach has remarkable performance compared to the basic greedy search and other advanced methods [17,18,49]. As discussed, Simulated Annealing (SA) starts with an initial network structure and randomly picks an edge change from the set of possible edge changes to the network. The new structure is accepted if it improves the network score. However, if the new structure decreases the network score, it is accepted based on the parameter of the system, temperature, T . At the start of the process, when the initial temperature is high, a lot of changes are accepted, even if the score is not improved. As the temperature decreases, less changes are accepted. This is continued until the desired stopping criterion is met. In Simulated Annealing with re-annealing (SAR), the temperature is allowed to rise in later parts of the search. This enhances the search to escape from the trap of local optima while thoroughly exploring the search space.

The more similar the inferred networks are to the reference networks, the better the search method [50]. If a regulatory relationship exists between two variables in two different networks, then that relationship, indicated by a link between the variables in both networks, is said to be common to the networks. If a link exists in both the network inferred by the methods and the reference network, it is considered as a true positive (TP). An inferred link is a false positive (FP) if it does not exist in the reference network. If a link exists in the reference network but not the inferred network by the methods, then it is considered as a false negative (FN). If a link is absent in both the reference network and the network inferred by the methods, it is said to be a true negative (TN). These indicators enable us to evaluate the performances of the algorithms in terms of sensitivity and specificity as in Eqs. (8) and (9).

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (8)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (9)$$

The higher the sensitivity and specificity values, the better the search algorithm. It is the predicted relationships by the methods that are examined to evaluate performance. These measures are appropriate for examining the performance of algorithms for inferring transcriptional regulatory networks as used in [47,50]. These well known measures, applied for such purposes, are important

since the search methods identify relationships for validation as illustrated [50,51]. Bayesian Network Inference with Java Objects (Banjo) [52] was used to implement SAGA, SAR GR, GS and SA as well as undertake quantile discretization of all variables of the datasets into 5 states. The temperature parameter setting was set to 10,000 with a 0.7 cooling factor. The BDe metric with an equivalent sample size of one is used in scoring the networks for selection as the search progresses. The experiments to obtain the sensitivity and specificity values were designed to terminate at any of the following: after 10,000 restarts or fixed times of 5 h, 2 h and 90 min. In all experiments, a minimum number of 1000 networks were explored before testing any of the stopping criteria.

3.4. Performance on human breast cancer dataset

The performances of the search methods for the breast cancer dataset consisting of 2453 probe sets (genes) and 89 arrays are shown in [Fig. 2](#). The prior network of 82 relationships (see [Supplementary Table S7](#)) was used to guide the search using all the five methods. These prior relationships served as a starting solution from the predictions of GENIE3 for subsequent exploration by the search methods without any restriction of retaining any of them in the final solution. Thus all possible relationships had equal chances of being included in the final solution depending on whether they improved the score of the networks.

3.4.1. Sensitivity

The results show that GR recorded the highest sensitivity in the first experimental design ([Fig. 2](#)). This affirms the statement in Chickering and Heckerman that GR is able to produce good results [17,49]. Though SAGA was not the best performer in the first evaluation, it produced higher sensitivity than both SAR and SA. These indicate that for learning BNs within shorter times from datasets with higher number of variables to arrays ratios, GR and GS learn relationships with higher true recoveries given higher numbers of prior relationships. In a further experiment in which time was increased to 2 h with a smaller prior network of 30 relationships (see [Supplementary Table S8](#) with probe set indices), SAGA outperformed the GR and GS while the SAR had zero sensitivity ([Fig. 2](#)). Though GS did not outperform GR, it performed better than both SAR and SA which had sensitivities of zero. This reflects the capacity of SAGA to recover true relationships from variables in a dataset given a smaller prior size and longer time periods. On the other hand, both SAR and SA recovered no true relationships in this second experiment leading to zero sensitivities indicating low performance of SAR and SA when they are supplied with less prior relationships. These results further indicate that the size of the

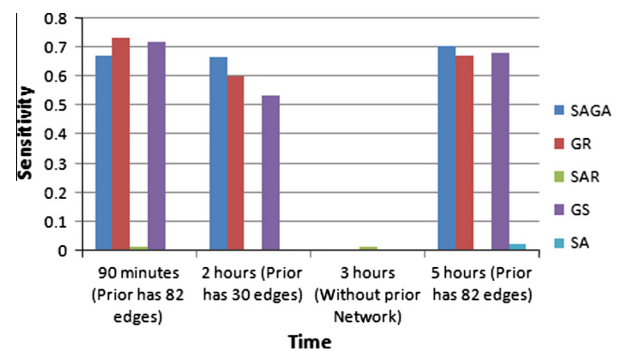


Fig. 2. Sensitivity comparison in human breast cancer dataset. SAGA has higher sensitivity than other search methods for problems involving priors. SAGA is Simulated Annealing with a Greedy Algorithm, GR is Greedy search with random restarts, SAR is Simulated Annealing with re-annealing, GS is Greedy Search and SA is Simulated Annealing.

prior network relationships supplied affect the true recovery rates of the search algorithms especially in the case of GR, GS, SA and SAR. In the third experimental design without prior network on this dataset, only SAR recovered some true relationships with less than 10% of sensitivity (Fig. 2). These results from experiments without priors indicate that inference with these methods is enhanced with some amount of known prior relationships among the variables. To further assess performance of algorithms over longer periods, the network of 82 prior relationships (see Supplementary Table S7) were used to guide the search methods to learn BNs over 5 h. SAGA's superior performance is reflected in higher true recoveries as confirmed by its high sensitivity values (Fig. 2). More importantly, the highest sensitivity for SAGA was obtained in this experimental design, indicating that it performs well in learning BNs from datasets with higher number of variables to arrays ratios over longer times. The zero sensitivity of the SAR run further indicates that increasing the time of search for learning BNs with SAR does not guarantee improvement in the true recoveries of results [20].

3.4.2. Specificity

Experimental designs similar to the ones in the previous section were carried out to find the specificities associated with all the methods. Generally, SAGA, GR and GS recorded high specificity scores, whereas SAR and SA had lower specificities for all experiments (Fig. 3). In particular, GS performed well in shorter runs of the experiment, recording the highest specificity in the first experimental set up. These indicate that for learning BNs within shorter times from datasets with high number of variables to arrays ratios, GR is able to distinguish false relationships from true relationships given a larger network of prior relationships. In the second experiment, SAGA recorded the highest specificity among all the search methods (Fig. 3). This further confirms SAGA's ability to distinguish false relationships from true relationships (given smaller networks of prior relationships) over the other BN search methods. Furthermore, these results suggest that all the BN search methods are affected by the number of prior relationships involved. SAGA was superior in learning relationships with BNs, and in distinguishing false relationships from true relationships, recording the highest specificity in the fourth experimental design (Fig. 3). Thus, SAGA performs very well in learning BNs from datasets with higher variables to arrays ratios given relevant prior network and sufficiently long time. Both SA and SAR recorded lower specificity values for all the experiments indicating that they are not ideal for learning BNs from datasets of this description.

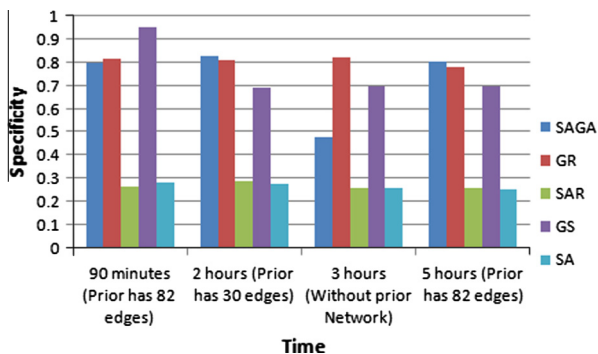


Fig. 3. Specificity comparison in human breast cancer dataset. SAGA competes well with existing algorithms as it records higher specificities in all the three experimental designs. SAGA is Simulated Annealing with a Greedy Algorithm, GR is Greedy search with random restarts, SAR is Simulated Annealing with re-annealing, GS is Greedy Search and SA is Simulated Annealing.

3.5. Performance on human airway smooth muscle dataset

Similar experiments as carried out on the human breast cancer data were conducted on the ASM dataset with the prior network of 82 relationships (see Supplementary Table S9) at 90 min and 5 h. Also, as was done in the breast cancer dataset at the 2-h time point, a reduced prior network of 30 relationships (see Supplementary Table S10) was used. The third experimental set-up was conducted without a prior network. This dataset had 168 probe sets and 54 arrays.

3.5.1. Sensitivity

The results of sensitivity tests are presented in Fig. 4. SAGA recorded the highest sensitivity for all the experimental designs except the third setup without priors (Fig. 4). The highest sensitivity was recorded at the second experimental design where prior relationships were reduced, confirming the observations of the breast cancer dataset. These observations confirm the superiority of SAGA to recover true relationships even in this smaller dataset. Though all the other methods did not perform as SAGA, they can recover a number of true relationships with non-zero sensitivities.

3.5.2. Specificity

Generally, all methods recorded high specificity scores attaining close to or more than 90% specificities (Fig. 5). In particular, GR recorded the highest specificities in all the experimental set-ups. Without loss of generality, it could be inferred that all the methods are able to truly distinguish false relationships from true relationships owing to their high performances on this smaller dataset. It further indicates that effective BN structure learning can be readily achieved with smaller datasets.

3.6. Performance on mouse whole brain dataset

Experimental designs carried out on the datasets in previous sections were also carried out on the mouse whole brain dataset with the prior network of 82 relationships (see Supplementary Table S11) and then a reduced prior network of 30 relationships (see Supplementary Table S12). The third design was constructed without prior network. This dataset consisted of 2797 probe sets and 411 arrays.

3.6.1. Sensitivity

SAGA recorded the highest sensitivity in all the experiments conducted with this large dataset except on the third design where none of the methods recorded any sensitivity (Fig. 6). SAGA performed just as GR at the 90 min using a prior network size of 82.

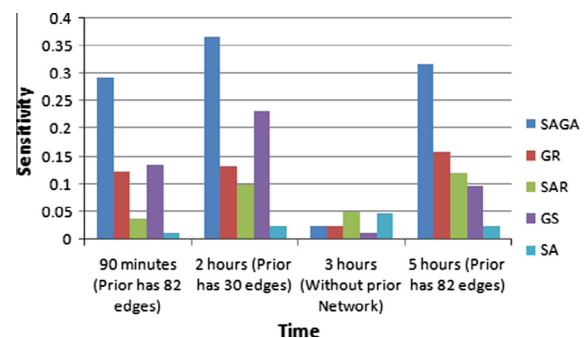


Fig. 4. Sensitivity comparison in the human airway smooth muscle dataset. SAGA identifies higher true transcriptional regulatory relationships for all experimental designs with prior network. SAGA is Simulated Annealing with a Greedy Algorithm, GR is Greedy search with random restarts, SAR is Simulated Annealing with re-annealing, GS is Greedy Search and SA is Simulated Annealing.

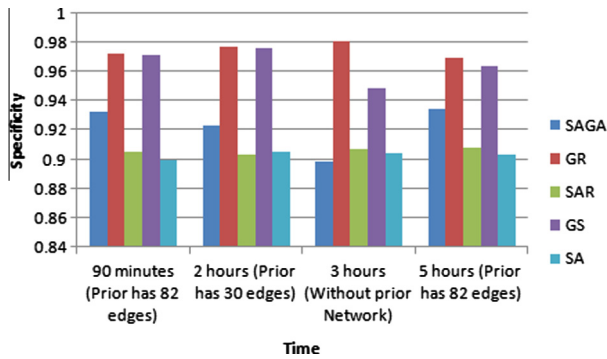


Fig. 5. Specificity comparison in the human airway smooth muscle dataset. All methods are able to distinguish false relationships. SAGA is Simulated Annealing with a Greedy Algorithm, GR is Greedy search with random restarts, SAR is Simulated Annealing with re-annealing, GS is Greedy Search and SA is Simulated Annealing.

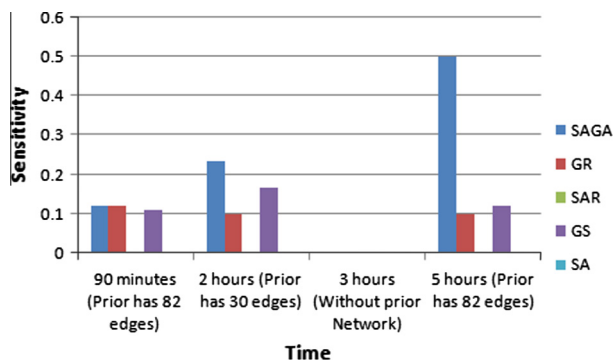


Fig. 6. Sensitivity comparison mouse whole brain dataset. SAGA is able to discover higher true relationships for all the experimental designs with prior networks in large datasets. SAGA is Simulated Annealing with a Greedy Algorithm, GR is Greedy search with random restarts, SAR is Simulated Annealing with re-annealing, GS is Greedy Search and SA is Simulated Annealing.

The highest sensitivity of SAGA was recorded in the fourth experimental design where the maximum time of 5 h were allowed (Fig. 6). The fact that the second experimental set up differed from sensitivities in the other set-ups, shows the size of the network of prior relationships affected the sensitivity of all the methods. In particular, smaller prior network enhanced the sensitivity of SAGA as it had the highest values in all the datasets for all the second experimental designs (i.e. at 2 h). The highest sensitivity recorded by SAGA in this set-up further confirms the higher true recovery rates as observed in the previous datasets. Though the performances of GR and GS were not better than SAGA, they were higher than SAR and SA which recorded no sensitivities for all the experimental designs for this dataset. These observations affirm that SAR and SA have lower sensitivities on larger datasets.

3.6.2. Specificity

In the first set-up, where all algorithms were permitted up to 90 min to infer regulatory relationships, GR demonstrated superior performance to clearly distinguish false discoveries from true discoveries by achieving the highest specificity (Fig. 7). Though all algorithms did not attain same specificities for this dataset, SAGA had comparable specificity to GR at 90 min.

3.7. Summary

SAGA recorded the highest sensitivities in the breast cancer and mouse whole brain datasets in the second and last experiments

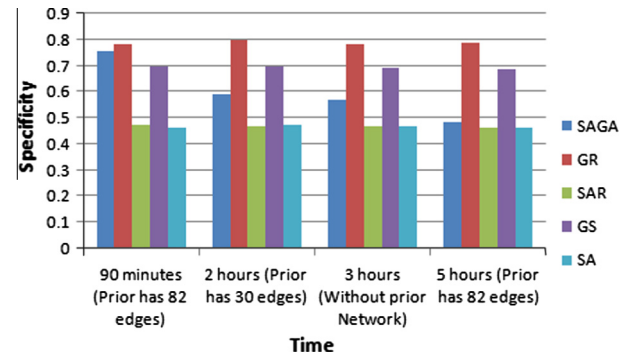


Fig. 7. Specificity comparison in mouse whole brain dataset. All methods are able to distinguish false discoveries from true relationships. SAGA is Simulated Annealing with a Greedy Algorithm, GR is Greedy search with random restarts, SAR is Simulated Annealing with re-annealing, GS is Greedy Search and SA is Simulated Annealing.

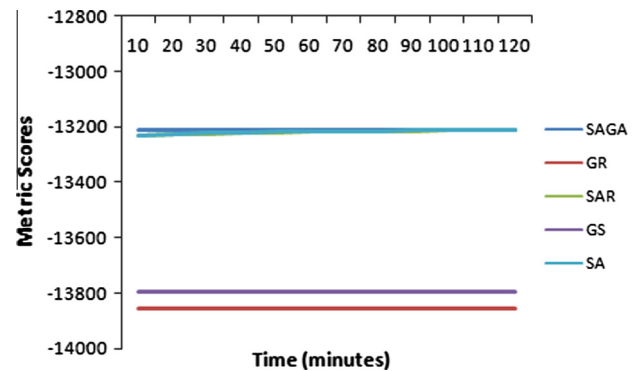


Fig. 8. Evolution of Performance metrics in the human airway smooth muscle dataset without prior network. Each experiment is repeated 10 times and the average is used. SAGA is Simulated Annealing with a Greedy Algorithm, GR is Greedy search with random restarts, SAR is Simulated Annealing with re-annealing, GS is Greedy Search and SA is Simulated Annealing.

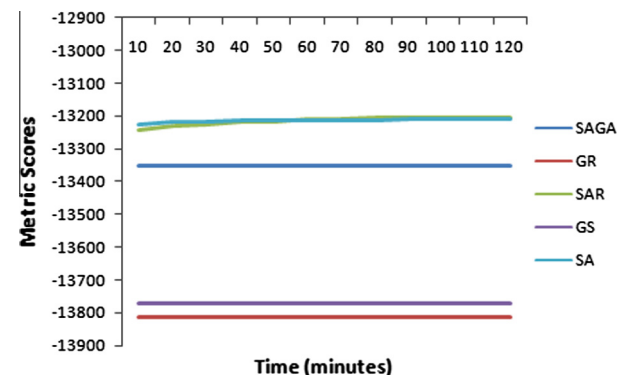


Fig. 9. Evolution of performance metrics in the human airway smooth muscle dataset with prior network with 30 edges. Each experiment is repeated 10 times and the average is used. SAGA is Simulated Annealing with a Greedy Algorithm, GR is Greedy search with random restarts, SAR is Simulated Annealing with re-annealing, GS is Greedy Search and SA is Simulated Annealing.

where smaller and larger prior relationships respectively were supplied. The performance of all the algorithms varied with respect to each dataset. In particular, for larger datasets where numbers of variables are large, SAGA is able to infer models with high rates of truly recovered relationships when used to infer BNs over longer periods with larger prior relationships. Therefore, SAGA is recommended for reverse-engineering networks from datasets where

Table 1
Results of all experiments.

Dataset	Experimental design	SAGA		GR		SAR		GS		SA	
		Se	Sp	Se	Sp	Se	Sp	Se	Sp	Se	Sp
Breast cancer	Priors (82 edges) ^a	0.671	0.796	0.732	0.815	0.012	0.266	0.720	0.952	0.000	0.281
	Priors (30 edges)	0.667	0.830	0.600	0.813	0.000	0.291	0.533	0.694	0.000	0.277
	Without prior	0.000	0.477	0.000	0.820	0.012	0.260	0.000	0.699	0.000	0.258
	Prior (82 edges) ^b	0.707	0.804	0.671	0.778	0.000	0.262	0.683	0.696	0.024	0.252
Human airway smooth muscle	Priors (82 edges) ^a	0.293	0.933	0.122	0.973	0.037	0.905	0.134	0.972	0.012	0.900
	Priors (30 edges)	0.367	0.923	0.133	0.977	0.100	0.903	0.233	0.976	0.024	0.905
	Without prior	0.024	0.899	0.024	0.981	0.049	0.907	0.012	0.949	0.048	0.904
	Prior (82 edges) ^b	0.317	0.935	0.159	0.970	0.120	0.908	0.098	0.964	0.024	0.903
Mouse whole brain	Priors (82 edges) ^a	0.122	0.757	0.122	0.782	0.000	0.473	0.110	0.695	0.000	0.463
	Priors (30 edges)	0.233	0.591	0.100	0.798	0.000	0.469	0.167	0.697	0.000	0.472
	Without prior	0.000	0.571	0.000	0.785	0.000	0.469	0.000	0.690	0.000	0.468
	Prior (82 edges) ^b	0.500	0.482	0.098	0.787	0.000	0.464	0.122	0.687	0.000	0.463

Se is Sensitivity; Sp is Specificity.

^a Experiment allowed for 90 min.

^b Experiment allowed for 5 h. SAGA is Simulated Annealing with a Greedy Algorithm, GR is Greedy search with random restarts, SAR is Simulated Annealing with re-annealing, GS is Greedy Search and SA is Simulated Annealing.

there are large numbers of variables, and where some priors are available. Moreover, GR has greater capacity to clearly distinguish false relationships from true relationships than SAGA. Thus, GR may be used to infer models which clearly distinguish false relationships from true relationships (for datasets where there are large numbers of variables, and where some prior knowledge is available).

On a smaller dataset as observed in the human airway smooth muscle dataset, SAGA is able to recover higher true positives since it had the highest sensitivities for all the experiments with priors. However, GR and GS are useful for such smaller datasets because of their higher specificity values on this dataset. In the absence of priors, all methods had low or no sensitivity values. This indicates that these methods are most effective in the presence of prior knowledge.

3.8. Evolution of performances

We further investigate the performance metrics for each algorithm. The scores of the metrics are only used to determine which neighbor (solution candidate) or network should be explored in subsequent explorations by the BN algorithms. The experiments were repeated 10 times and the average scores are used. In the experiments involving the ASM dataset without any prior network, SAGA showed comparatively high metric scores along with SA and SAR as time increased though it began with the highest scores (Fig. 8). These search methods produced candidate networks with higher scores than those generated by GS and GR (Fig. 8).

Further experiments involving the ASM dataset with prior networks of 30 edges and 82 edges, both SAR and SA evolved higher candidate networks with increasing time than SAGA, GR and GS (Figs. 9 and S1). These results indicate that SA and SAR perform higher exploration to search for local optima which are explored in consequent operations. The evolution of performance metrics suggest that greater metric scores by search methods do not guarantee higher sensitivities and specificities but provide candidate networks with higher scores for the subsequent explorations to derive networks of transcriptional regulatory networks. This is further supported by the experiments performed on the other datasets whose results are presented in Figs. S2, S3 and S4. Furthermore, both SA and SAR compromise on the sensitivities and specificities of the resultant networks despite generating candidate networks with higher metric scores. SAGA, which achieved intermediate metric scores among the search methods, did not

compromise on the sensitivities and specificities on this dataset (Figs. 4 and 5). Sensitivity and specificity are desirable to measure performances of algorithms for identifying transcriptional regulatory relationships among genes since they measure how well the algorithms uncover relationships from datasets. The sensitivities and specificities obtained from all the experiments are presented in Table 1.

4. Conclusion

We propose a sequential-based hybrid search method, SAGA for BN structure learning. SAGA uses known prior relationships to infer transcriptional regulatory relationships. The results of the evaluations of BN structure learning algorithms confirm that structure learning in BNs is effectively done with smaller datasets. However, when presented with real world problems, with prior knowledge reflecting prior relationships and large datasets of high ratios of variables to arrays, then SAGA will generally provide superior BN structure learning with high sensitivities and specificities. This is largely demonstrated by the results of the experiments performed on the datasets employed in this study. Therefore, in realistic contexts, identification of transcriptional regulatory networks could be enhanced with BN structure learning with SAGA.

Acknowledgments

This work was supported by a Novartis Institute for Biomedical Research/Ghana Biomedical Research Network fellowship. It was also supported by resources of MCPHS University, USA and the Kwame Nkrumah University of Science and Technology, Kumasi, Ghana.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at <http://dx.doi.org/10.1016/j.jbi.2014.08.010>.

References

- [1] Friedman N. Inferring cellular networks using probabilistic graphical models. *Science* 2004;303:799–805.
- [2] Sachs K, Perez O, Pe'er D, Lauffenburger DA, Nolan GP. Causal protein-signaling networks derived from multiparameter single-cell data. *Science* 2005;308:523–9.
- [3] Beaumont MA, Rannala B. The Bayesian revolution in genetics. *Nat Rev Genet* 2004;5:251–61.

- [4] Friedman N, Murphy K, Russell S. Learning the structure of dynamic probabilistic networks. In: Cooper G, Moral S, editors. Proceedings of the fourteenth conference on uncertainty in artificial intelligence (1998). San Francisco, CA: Morgan Kaufmann Publishers; 1995. p. 139–47.
- [5] Gardner T, di Bernardo D, Lorenz D, Collins J. Inferring Genetic Networks and identifying compound mode of action via expression profiling. *Science* 2003;301:102–5.
- [6] Gregoret F, Belcastro V, di Bernardo D, et al. A parallel implementation of the network identification by multiple regression (NIR) algorithm to reverse-engineer regulatory gene networks. *PLoS One* 2010;5(4):e10179.
- [7] 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.
- [8] Greenfield A, Madar A, Ostrer H, Bonneau R. DREAM4: combining genetic and dynamic information to identify biological networks and dynamical models. *PLoS One* 2010;5(10):e13397. <http://dx.doi.org/10.1371/journal.pone.0013397>.
- [9] Bansal M, Belcastro V, Ambesi-Impioabato A, di Bernardo D. How to infer gene networks from expression profiles. *Mol. Syst. Biol.* 2007;3:78. <http://dx.doi.org/10.1038/msb4100120>.
- [10] Chickering DM. Learning Bayesian network is NP-complete. *AI & STAT V*; 1996.
- [11] Chickering DM, Geiger D, Heckerman D. Learning Bayesian networks is NP-hard. Technical report MSR-TR-94-17, Microsoft Research; 1994. <<http://citeseerx.ist.psu.edu/viewdoc/versions?doi=10.1.1.49.7345>> [09.07.14].
- [12] Robinson RW. Counting unlabeled acyclic digraphs. In: Little CHC, editor. Combinatorial mathematics. Lecture notes in mathematics, vol. 622. Australia: Springer-Verlag; 1997.
- [13] Friedman N, Nachman I, Pe'er D. Learning Bayesian network structure from massive datasets: the sparse candidate algorithm; 1999. <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.37.5605>> [24.06.13, date last accessed].
- [14] Nielsen J, Kocka T, Pena J. On local optima in learning Bayesian Networks. In: Proceedings of the nineteenth conference on uncertainty in artificial intelligence (2003). San Francisco, CA: Morgan Kaufmann Publishers; 2003. p. 435–42.
- [15] Gamez JA, Puerta JM. Constrained score + (local) search methods for learning Bayesian Networks. In: Symbolic and quantitative approaches to reasoning with uncertainty: lecture notes in computer science, vol. 3571. Heidelberg: Springer-Verlag; 2005. p. 161–73.
- [16] Yuan C, Malone B, Wu X. Learning optimal Bayesian Networks using A* search. In: Proceedings of the twenty-second international joint conference on artificial intelligence, vol. 3. California: AAAI Press; 2011. p. 2186–91. <<http://ijcai.org/papers11/Papers/IJCAI11-364.pdf>> [20.06.13, date accessed].
- [17] Chickering DM. Learning equivalence classes of Bayesian Network structures. In: Horvitz E, Jensen FV, editors. Proceedings of the twelfth annual conference on uncertainty in artificial intelligence. USA: Oregon; 1996. p. 150–7.
- [18] Gamez JA, Mateo JL, Puerta JM. Learning Bayesian networks by hill climbing: efficient methods based on progressive restriction of the neighborhood. *Data Min Knowl Disc* 2011;22:106–48.
- [19] Heckerman D. A Bayesian approach to learning causal networks. In: Proceedings of eleventh conference on uncertainty in artificial intelligence. San Francisco, CA: Morgan Kaufmann Publishers; 1995. p. 285–95.
- [20] Janzura M, Nielson JA. Simulated annealing-based method for learning Bayesian Networks from statistical data. *Int J Intell Syst* 2006;21:335–48.
- [21] Jaakkola T, Sontag D, Globerson A, et al. Learning Bayesian Network structure using LP relaxations. In: Proceedings of the 13th international conference on artificial intelligence and statistics (AISTATS): 13–15 May 2010, Sardinia, Italy, vol. 9 of JMLR: W&CP 9; 2010. <<http://jmlr.org/proceedings/papers/v9/>> [20.06.13, last date accessed].
- [22] Wang M, Chen Z, Cloutier S. A hybrid Bayesian network learning method for constructing gene networks. *Comput Biol Chem* 2007;31(5–6):361–72.
- [23] de Campos LM, Fernandez-Luna JM, Gamez JA, Puerta JM. Hill climbing optimization for learning Bayesian networks. *Int J Approximate Reasoning* 2002;31:291–311.
- [24] Daly R, Shen Q. Learning Bayesian Network equivalence classes with ant colony optimization. *J Artif Intell Res* 2009;35:391–447.
- [25] Larrañaga P, Karshenas H, Bielza C, Santana R. Reviews evolutionary algorithms in Bayesian Network learning task. *Inf Sci* 2013;233:109–25. <http://dx.doi.org/10.1016/j.ins.2012.12.051>.
- [26] Larranaga P, Poza M, Yurramendi Y, Murga RH, Kuijpers CMH. Structure learning of Bayesian Networks of genetic algorithms: a performance analysis of control parameters. *IEEE Trans Patterns Anal Mach Intell* 1996;18(9):912–26.
- [27] Larranaga P, Murga R, Poza M, Kuijpers C. Structure learning of Bayesian Networks by hybrid genetic algorithms. *Lect Note Stat* 1996;112:165–74.
- [28] Lee J, Chung W, Kim E, Kim S. A new genetic approach for structure learning of Bayesian networks: matrix genetic algorithm. *Int J Control Autom Syst* 2010;8(2):398–407.
- [29] Cowie J, Oteniya L, Coles R. Particle swarm optimization for learning Bayesian Networks. In: Proceedings of the world congress on engineering 2007, vol. I WCE 2007, July 2–4, 2007, London, UK.
- [30] Sivanandam SN, Deepa SN. Introduction to genetic algorithm. New York: Springer-Verlag Berlin Heidelberg; 2008.
- [31] Adabor ES, Acquah-Mensah GK, Oduro FT. Inferring transcriptional regulatory relationships among genes in breast cancer: an application of Bayes' theorem. *Int J Stat Probab* 2014;3(2):52–62. <http://dx.doi.org/10.5539/ijsp.v3n2p52>.
- [32] Pearl J, Verma T. A theory of inferred causation. In: Allen J, Fikes R, Sandewall E, editors. Principles of knowledge representation and reasoning: proceeding of the second international conference. San Mateo, CA: Morgan Kaufmann; 1991. p. 441–52.
- [33] Spirtes P, Glymour C, Scheines R. Causation, prediction, and search, 2nd ed.. Cambridge, MA: MIT Press; 2000.
- [34] Dash D, Druzdzel MJ. A hybrid anytime algorithm for the construction of causal models from sparse data. In: Laskey KB, Prade H, editors. Proceedings of the sixth annual conference on uncertainty in artificial intelligence (UAI'99), July 30–August 1 1999, Stockholm Sweden. Morgan Kaufman; 1999. p. 142–9.
- [35] Acid S, de Campos LM. A hybrid methodology for learning belief networks: BENEDICT. *Int J Approx Reason* 2001;27(3):235–62.
- [36] Tsamardinos I, Brown LE, Aliferis CF. The max–min Hill climbing Bayesian network structure learning algorithm. *Mach Learn* 2006;65:31–78. <http://dx.doi.org/10.1007/s10994-006-6889-7>.
- [37] Cooper GF, Herskovits E. A Bayesian method for the induction of probabilistic networks from data. *J Mach Learn* 1992;9:309–47.
- [38] Heckerman D, Geiger D, Chickering DM. Learning Bayesian networks: the combination of knowledge and statistical data. *J Mach Learn* 1995;20(3):197–243.
- [39] Klebaner FC. Introduction to stochastic calculus with applications, 2nd ed.. UK: Imperial College Press; 2005.
- [40] Greenfield A, Hafemeister C, Bonneau R. Robust data-driven incorporation of prior knowledge into the inference of dynamic regulatory networks. *Bioinformatics* 2013;29(8):1060–7.
- [41] Marbach D, Prill RJ, Schaffter T, et al. Revealing strengths and weaknesses of methods for gene network inference. *Proc Natl Acad Sci* 2010;107(14):6286–91.
- [42] Stolovitzky G, Monroe D, Califano A. Dialogue on reverse-engineering assessment and methods: the DREAM of high-throughput pathway inference. *Ann N Y Acad Sci* 2007;1115:1–22.
- [43] Irizarry RA, Hobbs B, Collin F, et al. Exploration, normalization and summaries of high density oligonucleotide array probe level data. *J Biostat* 2003;4(2):249–64.
- [44] Misior AM, Deshpande DA, Loza MJ, Pascual RM, Hipp JD, Penn RB. Glucocorticoid- and protein kinase A-dependent transcriptome regulation in airway smooth muscle. *Am J Respir Cell Mol Biol* 2009;41:24–39.
- [45] Bhavé SV, Hornbaker C, Phang TL, Saba L, Lapadat R, Kechris K, et al. The PhenoGen informatics website: tools for analyses of complex traits. *BMC Genet* 2007;8:59.
- [46] Acquah-Mensah GK, Taylor RC, Bhavé SV. PACAP interactions in the mouse brain: implications for behavioral and other disorders. *Gene* 2012;491:224–31.
- [47] Huynh-Thu VA, Irtuthum A, Wehenkel L, Geurts P. Inferring regulatory networks from expression data using tree-based methods. *PLoS One* 2010;5(9):e12776.
- [48] Faith JJ, Hayete B, Thaden JT, Mogno I, Wierzbowski J, Cottarel G. Large-scale mapping and validation of *Escherichia coli* transcriptional regulation from a compendium of expression profile. *PLoS Biol* 2007;5(1):e8. <http://dx.doi.org/10.1371/journal.pbio.0050008>.
- [49] Chickering DM, Heckerman D. Efficient approximation for the marginal likelihood of Bayesian networks with hidden variables. *J Mach Learn* 1997;29:181–212.
- [50] Husmeier D. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *J Bioinform* 2003;19(17):2271–82.
- [51] van Berlo RJP, van Someren EP, Reinders MJT. Studying the conditions for learning dynamic and Bayesian Networks to discover genetic regulatory networks. *Simulation* 2003;79(12):689–702.
- [52] Sladeczek J, Hartemink AJ, Robinson J. Banjo. <<http://www.cs.duke.edu/~amink/software/banjo/>> [20.06.13, date last accessed].