JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 24, Number 4, 2017
Mary Ann Liebert, Inc.
Pp. 340–356
DOI: 10.1089/cmb.2016.0100

New Algorithm and Software (BNOmics) for Inferring and Visualizing Bayesian Networks from Heterogeneous Big Biological and Genetic Data

GRIGORIY GOGOSHIN,[1] ERIC BOERWINKLE,[2,3] and ANDREI S. RODIN[1]

ABSTRACT

Bayesian network (BN) reconstruction is a prototypical systems biology data analysis approach that has been successfully used to reverse engineer and model networks reflecting different layers of biological organization (ranging from genetic to epigenetic to cellular pathway to metabolomic). It is especially relevant in the context of modern (ongoing and prospective) studies that generate heterogeneous high-throughput omics datasets. However, there are both theoretical and practical obstacles to the seamless application of BN modeling to such big data, including computational inefficiency of optimal BN structure search algorithms, ambiguity in data discretization, mixing data types, imputation and validation, and, in general, limited scalability in both reconstruction and visualization of BNs. To overcome these and other obstacles, we present BNOmics, an improved algorithm and software toolkit for inferring and analyzing BNs from omics datasets. BNOmics aims at comprehensive systems biology—type data exploration, including both generating new biological hypothesis and testing and validating the existing ones. Novel aspects of the algorithm center around increasing scalability and applicability to varying data types (with different explicit and implicit distributional assumptions) within the same analysis framework. An output and visualization interface to widely available graph-rendering software is also included. Three diverse applications are detailed. BNOmics was originally developed in the context of genetic epidemiology data and is being continuously optimized to keep pace with the ever-increasing inflow of available large-scale omics datasets. As such, the software scalability and usability on the less than exotic computer hardware are a priority, as well as the applicability of the algorithm and software to the heterogeneous datasets containing many data types—single-nucleotide polymorphisms and other genetic/epigenetic/transcriptome variables, metabolite levels, epidemiological variables, endpoints, and phenotypes, etc.

Keywords: Bayesian network(s), big data, omic data, systems biology.

[1]Diabetes and Metabolism Research Institute, City of Hope, Duarte, California.
[2]Human Genetics Center, School of Public Health, University of Texas Health Science Center, Houston, Texas.
[3]Institute of Molecular Medicine, University of Texas Health Science Center, Houston, Texas.

1. BACKGROUND

Continuous progress in the development of high-throughput genotyping and sequencing technologies led to the information overload problem that is likely to get exacerbated as the tools become even more accessible. Translated into the data analysis vernacular, the challenge is essentially threefold: (1) increasing the scalability of the data analysis methodology to accommodate large-scale datasets and (2) incorporating many heterogeneous data types into the analysis framework, while (3) being able to account for, and interpret, nonadditive interactions between the variables (genetic and otherwise). These three components are closely interrelated, in that trying to build more complex and expressive models (containing many data types and allowing for high-order variable interactions) necessarily and severely compromises scalability. Indeed, baseline analysis of a typical genetic epidemiology dataset, for example, one million univariate single-nucleotide polymorphism (SNP)–phenotype tests corrected for multiple comparisons, generated by a genome-wide association study (GWAS) is computationally undemanding, but does not even begin to address the latter two issues, while more sophisticated analysis methods tend to be NP-hard and therefore computationally infeasible for the datasets containing large numbers of variables.

Many of such methods belong to the domain of systems biology data analysis. Their primary goal is to grasp the underlying biological system in its entirety, including the high-order interactions between the variables contained in the data, potentially leading to the generation of new biological hypotheses of varying levels of complexity. Coincidentally, additional challenges arise in (1) presenting the output of the systems biology data analysis method in a format that can be understood and interpreted by a human expert in a specific biomedical research field (e.g., genetic epidemiology of particular trait/disease in case of GWAS genotype–phenotype mapping) and (2) guarding against overfitting/overparameterization (learning spurious relationships from the datasets of unfavorable dimensionality, that is, the too many variables—too few observations problem).

A prototypical systems biology data analysis method is Bayesian network (BN) modeling, in which a graphical model (BN) of joint probability distribution of random variables contained in the dataset is reconstructed directly from the data. All observed biological variables and measurements are understood to have a probabilistic nature. Thus, in BNs, nodes correspond to random variables and edges to dependencies. The edges are directed, reflecting (sometimes arbitrary, strictly for the purposes of mathematical convenience) ancestor–descendant relationships between the variables. This directionality is important as it defines a unique representation for the multiplicative partitioning of the joint probability and, subsequently, a direction of inference in the BN once a BN structure (topology) is inferred. The absence of an edge between the two variables indicates conditional independence (although this cannot be strictly guaranteed in practice).

BN modeling, and network implementation in general, has been extensively used in genetics, bioinformatics, and computational biology since the turn of the century (Friedman et al., 2000; Pe'er, 2005; Rodin et al., 2005; Djebbari and Quackenbush, 2008; Rodin et al., 2012; Liu et al., 2014; Lo et al., 2015; Tasaki et al., 2015; Li et al., 2016, to name but a few). A detailed treatment of BN methodology, while outside of the scope of this communication, can be found in Pearl (1988), Pearl (2000), Heckerman (1995), and Chickering et al. (2004). We also refer the reader to Rodin et al. (2012) and references therein for interpretation of ancestor–descendant relationships, causality, directionality, and BN validation in the genetic epidemiology context (defined by datasets containing many SNP variables and phenotype and epidemiological variables, among others). Briefly, the advantages of the BN modeling over simpler, less expressive, data analysis methods are (1) the ability to incorporate many different data types into analysis, (2) the ability to account for high-order variable interactions (e.g., epistatic and gene environment), and (3) an output (a graphical network model) that is easily understood and interpreted by a human expert. In addition, unlike certain other methods operating in Euclidean space, the BN approach is context independent and has a number of attractive theoretical properties allowing mixing of different data types in a theoretically sound probabilistic framework.

In addition, due to the natural sparseness of biological systems (i.e., each node in a network being directly connected with a limited number of other nodes), the resulting BN models are relatively easy to compartmentalize, which augurs well for both reconstruction and visualization scalability. Thus, BNs are an excellent biological modeling and hypothesis generation tool. Three major practical difficulties associated with BN modeling in our research context are (1) limited scalability, at least compared with the more simplistic analyses, (2) absence of the readily available software aimed primarily at the biomedical data

that can take advantage of the known genetic data type structures and various omics formats, and (3) interpretation, validation (e.g., using statistical resampling), and visualization of larger BNs. A specific, but persistent, challenge, on both theoretical and practical levels, is combining continuous and discrete variables together into a comprehensive hybrid probability model within the same BNs.

2. EXISTING ALGORITHMS AND SOFTWARE PACKAGES

Two useful BN reconstruction software lists can be found at (http://www.cs.ubc.ca/~murphyk/Software/bnsoft.html and http://www.kdnuggets.com/software/bayesian.html). These include both commercial and free general-purpose BN modeling software and BN-based classifiers. A more selective list of free packages, compared in the bioinformatics application context, is compiled in Paluszewski and Hamelryck (2010), with a special emphasis on the dynamic BNs (DBNs). pwOmics (Watcher and Beisbarth, 2015) is the most recent implementation of DBN modeling in omics data context. Another recent BN reconstruction algorithm (Jiang et al., 2010a, 2010b; Jiang and Neapolitan, 2012) is of special interest as the authors attempted to increase the scalability of BN modeling to make it directly usable with the large-scale genetic epidemiology datasets. However, it has limitations related to the constraints imposed on the general BN structure and/or on the variables effectively preselected for BN analysis. This work had been followed up in Neapolitan et al. (2014), Jiang and Neapolitan (2015), and Jiang et al. (2015), but remains subject to these and related limitations. Large-scale genetic epidemiology dataset BN analysis was also pursued in Han et al. (2012) at the cost of specifying a single target variable. BN Webserver (Ziebarth et al., 2013) is a comprehensive biological BN analysis tool, which, among other things, efficiently deals with hybrid models (heterogeneous variables/data types) in a biological user-friendly manner; unfortunately, its scalability is essentially nonexistent (<20 nodes).

Constructing multilevel gene regulatory networks (Guan et al., 2014) aims at ChiP-seq and gene expression data, but suffers from the same major shortcoming (limited scalability). A more theoretically rigorous approach of effectively reducing BN to a Markov neighborhood of a variable of interest (Gao and Ji, 2016) is intriguing, but has not been deployed in actuality. Similarly, a theoretically attractive approach to inferring causality *via* intervention data (Cho et al., 2016) suffers, once again, from low scalability and limited deployment. In general, theoretical rigor and distributional flexibility on one hand and scalability on the other tend to be mutually exclusive (see Yin et al., 2015a, for another recent example).

As an important aside, when developing BNOmics, complete code transparency was a priority. This makes it much easier to change and augment the BN reconstruction engine (local search/optimization algorithm) on the fly. Therefore, BNOmics is explicitly designed to be sufficiently flexible to incorporate different variations of baseline search algorithms, network scoring functions, and discretization and imputation approaches. As such, BNOmics engine is ideally suited to be incorporated into a typical comparative simulation study framework. It should be emphasized that first and foremost, BNOmics is a prototype/proof-of-concept design of a research platform prioritizing simplicity, flexibility, and adaptability to various biomedical data analysis applications rather than an overly complex production-level software package with all imaginable options and extensions.

3. ALGORITHM AND IMPLEMENTATION

BNOmics is realized as a series of Python scripts, including the data formatting and storage facilities, actual BN reconstruction engine, output interface, and various optional support routines (data reformatting plug-ins). A Python interpreter with a standard set of modules as well as additional numerical libraries (numpy) is required to run the software. Help (readme) files and the example input data files (see section 4 for the example application) are provided as part of the distribution. Computationally, most intensive parts of BN reconstruction engine are implemented in C++ using ctypes interface.

3.1. Data storage and input format

The input data file is a plain, flat (variables by observations/individuals) text file in a format similar to the typical comma-delimited spreadsheet export file. Loading from other common file formats, streams, and strings is also supported. Because the basic BN reconstruction algorithm uses multinomial local probability

model, in the baseline implementation, discretization of continuous variables is necessary (but see section 3.2). Optional scripts are available for automated input file generation, including common discretization procedures (equal size bins, equal value ranges, entropy-based discretization, etc.). In the context of genetic epidemiology datasets, most variables are discrete by nature (e.g., SNPs, allelic states); however, one should be careful when discretizing continuous phenotypes or, for example, metabolomic measurements. Therefore, if possible, user-driven manual or semimanual discretization is advised (and can be easily accomplished on the fly within the Python environment—it is precisely the flexibility of such nature that led us to choose Python over other languages). Similarly, we advise carrying out user-driven missing value imputation before engaging the BNOmics software—although optional imputation routines (using majority, frequency, and proximity rules) are available, sensible imputation is highly dependent on the specific data type and quality control procedures implemented during the data generation stage.

For example, when analyzing metabolomic data, it is difficult to distinguish between the metabolite measurement value missing due to a technical error, low metabolite concentration, or the actual metabolite absence in the sample. Such technical artifacts have to be dealt with manually or semimanually, and with large datasets, the only practical way to do so is to algorithmically parse the data (which, again, is easily achieved by using a Python interpreter as a universal control interface). A more advanced imputation algorithm, based on the local probabilistic inference in the immediate network neighborhood of a variable in question (with missing values), is also available, but its properties have not been comprehensively assessed yet and it will be described in detail elsewhere (A potential concern with such approaches is their general susceptibility to overfitting.).

There is no explicit algorithmic or software limit imposed on the input dataset size (number of variables and observations/individuals); however, data files larger than 2–3 GB are not recommended for a typical workstation (16 GB memory or less) installation without certain modifications (to keep it in perspective, 500K variables-strong GWAS dataset containing $\sim$2000 case/controls fits in just under 1.5GB). To optimize the data storage, retrieval, and memory I/O access in these situations, the actual data (single-type variable values) have to be stored separately from the annotation file, row-wise, following the approach espoused by Nielsen and Mailund (2008) to compensate for disk-bound I/O latency. When dealing with extremely large data files, some form of batch data access may be required, with the whole segments of the file loaded directly into RAM (which explains the need for row-wise data structure). It should be noted that these issues are very architecture and problem specific and are best addressed on a case-by-case basis within Python interface.

Among the implemented speed/scalability improvement measures is the integer-type encoding and representation of the data in memory, which leads to a smaller footprint, a faster access, and an ability to directly apply a number of efficient numerical operations. The input dataset entries are coded in 0, 1, 2,… format. Further improvements are a result of an optimized algorithm design (section 3.3). Together, these measures increase computational efficiency by approximately an order or two of magnitude (depending on the data and compared with a typical BN software implementation) without imposing restrictions on BN structure. It remains to be mentioned that current limit on the number of samples is flexible and depends on hardware. In a recent application, for example, $\sim 10^7$ samples (by 100+ variables), dataset analysis (results not shown) was completed in under 24 hours. While the algorithm is, in principle, linear with respect to the number of samples, at some point it becomes a bottleneck (see section 4.2).

3.2. Continuous and discrete variables

In baseline implementation, due to the limitations of linear Gaussian (for continuous variables) and hybrid BN local probability models (Friedman et al., 2000; Pe'er, 2005), continuous variables in the dataset have to be discretized. This is customary not just for BN modeling but also for many other data analysis methods. However, there are no commonly agreed upon discretization guidelines or standards in bioinformatics, and existing research tends to be very microarray data-centric (Vass et al., 2011). This paucity can, in large part, be attributed to our limited understanding of the theoretical motivation behind the discretization. Indeed, the actual purpose of discretization is twofold: first, to come up with the partitioning (into bins or events) that best reflects characteristic features of the distribution of a continuous variable, and second, to preserve the intervariable relationships (correlation, dependency) and their relative strengths for subsequent data analysis by the algorithms or statistical tests developed for discrete variables. While these two goals obviously overlap, our primary interest lies with the latter and not the former. For example, in the

context of genetic epidemiology, we should aim to discretize the continuous variables, such as quantitative trait (QT) endpoints and intermediate phenotypes, in a manner that maintains true relative strengths of SNP-phenotype associations.

Consequently, our considerations are motivated by the fact that partitioning is equivalent to selecting only a few events from the sigma algebra of events associated with the distribution in question, and no selection is any more representative than any other when it comes to establishing conditional independence because it has to be established across all events. Another way of looking at it is while any Borel measurable (reasonable and deterministic) partitioning function is independence-preserving, *sensu stricto* independence of partitioned variables implies nothing with regard to the independence of original variables. On the other hand, to falsify the independence assumption, it is sufficient to have at least one dependent event, which, however, is not known *a priori*. Hence, the only reasonably efficient way to approach the problem is from the perspective of falsification of the independence assumption, which is significantly less time-consuming than verifying independence over all possible events.

In general, our principal proposition for manual or semimanual discretization is that the simplest discretization method (that satisfies basic common sense data type-specific requirements) should be chosen. While partitioning into the fewer bins potentially might be perceived as leading to increased information loss, this is, in fact, only an illusion generated by poor interpretability of often complicated conditional independence, which does not care about the finer features nearly as much as about the change in the conditional distribution over multiple scales (when conditioned on new variables). What matters the most is how the bins/events of one variable intersect/interact with bins/events of another variable or a set of variables. Moreover, any fears associated with coarser partitioning over the finer partitioning should be counterbalanced by the corresponding decrease in random noise and natural variation customary for the biological variable measurements. At the same time, it has been observed by us and others that partitioning into lower number of bins tends to lead to higher edge density in reconstructed BNs (Clarke and Barton, 2000; Rodin et al., 2005), possibly because BN scoring metrics are biased toward fewer variable values. Regardless, coarser binning can be considered as boosting sensitivity, while finer binning as boosting specificity. Technically, this is due to the fact that the conditional probability distributions with fewer states carry fewer constraints, which are therefore easier to satisfy when it comes to conditional entropy minimization (or conditional probability maximization).

Such moderate overfitting, especially in the Markov locality of the continuous variables (e.g., QT phenotype variables in GWAS datasets), is actually beneficial for the purposes of exploratory data analysis (automated hypothesis generation). Typically, discretizing into two or three bins with entropy-based partitioning points (Fayyad and Irani, 1993), as long as it does not explicitly conflict with the observed nature of the continuous variable, is to be preferred. It is also more favorable from the computational and memory utilization efficiency viewpoint. This is the default option in BNOmics (maximum entropy clustering-based discretization into a minimal number of bins). However, there is also a novel option of treating both continuous and discrete variables simultaneously. It is fully functional, but at this time does not scale up as well as purely discrete variable design (see section 4.3).

3.3. BN reconstruction algorithm

BN reconstruction is generally a two-stage process, involving model selection (search of the network structure or topology that best fits the data) and probabilistic inference propagation given the fixed network structure. The former is NP-hard and therefore some local or heuristic search algorithm is usually employed (instead of exhaustive search) for any dataset of nontrivial size (starting with $\sim$20 nodes/variables, exact algorithms and computations become infeasible). The candidate network structures, or models, are evaluated using an objective scoring function (metric). These often incorporate, explicitly or implicitly, a model complexity penalty to prevent overfitting. In addition, an initial search state (network structure prior) has to be specified—selection of an optimal (or biologically meaningful) prior has received much attention (e.g., Friedman et al., 1999; Steele et al., 2009; Keilwagen et al., 2010; Zhang et al., 2014), in part, because it may alleviate the scalability problem to a degree. The default BNOmics algorithm is completely agnostic in regard to the BN structure and prior and therefore is entirely data driven. However, restrictions on the network structure (i.e., forbidding or forcing edges between the network nodes) can easily be accommodated if desired.

Once the input data file is initialized, the BNOmics algorithm works as follows: given the data $D=\{x_1,\ldots, x_N\}$, where N is the number of variables, we aim to assign these variables to the N nodes of a BN (Fig. 1).

a

Procedure 1

Input: x_i, π_i
Output: $\arg\max_j F(x_i, \pi_i \cup x_j), \quad \max_j F(x_i, \pi_i \cup x_j) - F(x_i, \pi_i)$

```
j_max ← none
score ← F(x_i, π_i)
Δs ← 0
for all j ∈ {k ∈ I : x_k ∉ π_i, k ≠ i} do
  s ← F(x_i, π_i ∪ x_j)
  if s > score then
    score ← s
    Δs ← Δs + s − score
    j_max ← j
  end if
end for
return j_max, Δs
```

b

Procedure 2

Input: Data D
Output: Network structure S

```
for all i ∈ I do
  π_i ← ∅
  (p_i, Δs_i) ← Apply Procedure (1)
end for
while max_j{Δs_j : ∀j ∈ I} > 0 do
  i ← arg max_j{score_j : ∀j ∈ I}
  j ← p_i
  π_i ← π_i ∪ x_j
  P ← ancestors of j-th node
  C ← descendants of i-th node
  for all k ∈ P ∪ {j} do
    for all c ∈ C ∪ {i} do
      Forbid edge (c, p)
    end for
    (p_k, Δs_k) ← Reapply Procedure (1) to k-th node
  end for
end while
return S = {π_i : ∀i ∈ I}
```

FIG. 1. BN reconstruction algorithm kernel pseudocode (Procedures 1 and 2). BN, Bayesian network.

Let $I=\{1,\ldots,N\}$ be the index set of nodes and π_i be the parent set of the i-th node x_i. We also introduce a real-valued objective (scoring) function $F(x_i, \pi_i)$ that has two parameters, the node and its ancestor set. By changing the ancestor set, we can evaluate different subnetworks centered around the node. The simplest (first-order) operator would be an addition of a node x_j to the ancestor set π_i of the node x_i. By varying j over $\{k \in I : x_k \notin \pi_i, k \neq i\}$, we can score each configuration $s_{ij}=F(x_i, \pi_i \cup x_j)$. Procedure 1 (Fig. 1) details the process of finding the best single ancestor addition to a node. It returns j_{max}, the index of a node that (when added as an ancestor) maximizes the scoring function. Thus, it generates the best edge to be added to the network in the immediate Markov neighborhood of x_i (in its upper part—the parents of x_i), the edge representing a pair of nodes (i,j_{max}). We call this first-order search as the only allowable operator is adding a single edge. After the addition of an edge, a search for an optimal edge to remove from the ancestor set is performed. If such an edge is found, it is dropped and the necessary changes are propagated through the structure.

This combination of add and drop searches allows to maintain a relatively optimal ancestor set (the task of finding an optimal ancestor set is one of the fundamentally difficult problems, and a principal contributor to NP-hardness of network reconstruction, and as such is solved only approximately). Similarly, second-order search allows adding two new edges, evaluating $s_{ijk}=F(x_i, \pi_i \cup x_j \cup x_k)$, and so on. Such higher-order searches are obviously more computationally intensive, and the baseline BNOmics implementation uses

first-order search only, although stochastic second-order search has been implemented and tested (second-order search is quadratic $O(n^2)$ in the number of variables, and if $\sim n$ pairs of variables are sampled at random from $(n^{2-n})/2$ total pairs, the second-order search becomes O(n)).

It should be noted that first-order search is still capable of discovering higher-order interactions in the general sense as well as in the more specific biological sense (e.g., epistatic interactions between multiple SNPs in a Markov neighborhood of a phenotype node). Not only that, it is possible to arrive to the same ancestor set using different methods as the only thing that matters from the perspective of the order of accounted for interactions is the ancestor set itself. However, higher-order searches do bring us closer to the exhaustive search ideal (although in practice, they provide only marginal improvement, reflected in finer detail, over other search strategies).

Because no network structure (topology) priors are assumed, we start with the empty network; all ancestor sets are initialized to the empty sets, $\pi_i \leftarrow \varnothing$, $\forall i \in I$. Equivalently, we assume independence of all nodes and then try to falsify this assumption by establishing edges. Then, Procedure 1 is applied to each node to obtain a set of N candidate best edges *(i, j)* with associated scores (first loop in Procedure 2, Fig. 1). The second loop in Procedure 2 (Fig. 1) selects the highest scoring edge and rearranges the current state of the network accordingly (performs bookkeeping). In effect, Procedure 2 reconstructs the network topology by incremental edge addition (forward selection), creating a partial ordering of the nodes along the way. In that respect, it is an elaborate extension of the basic greedy search (hill-climbing) local search/optimization algorithm. It remains to be noted that different stopping criteria (in addition to the scoring function improvement-based one shown in Fig. 1) can be used; in practice, the second loop of Procedure 2 can be bound by the preset iteration limit (i.e., CPU time) or, for smaller networks, the search can continue until full exhaustion (defined by the largest possible increase of the scoring function being commeasurable with the machine epsilon/precision). In addition to the default gradient descent with constraints, stochastically perturbed restarts are also available as an option (to deal with local minima). Optionally, stochastically perturbed restarts are performed as stochastic relaxation over the structure and subsequent reconvergence until a more optimal structure is found.

This algorithm compares favorably with other recent attempts to scale up the BN reconstruction (see section 2 above; also Friedman et al., 1999), in that it does not limit the BN structure flexibility by treating the SNPs, gene expression measurements, or other subgroups of variables separately and testing them for pairwise correlation (implying, for example, linkage disequilibrium or first-order epistatic interaction in case of SNPs) before reconstructing the BN. This said, majority of BN reconstruction algorithms (Neapolitan, 2004), including the one detailed in this communication, reside somewhere in between basic greedy search and exhaustive search (much closer to the former for any sufficiently large datasets), and comprehensive simulation studies are necessary to ascertain their relative performance and robustness within different application domains.

The default scoring measure used by BNOmics is MDL/BIC K2 (Cooper and Herskowitz, 1992), a metric directly proportional to the probability of observing the data given the BN structure (i.e., marginal likelihood) that also penalizes explicitly for the model complexity (Schwarz, 1978). Alternatives include AIC (Akaike, 1974) and different flavors of MDL/BIC (Rissanen, 1978; Suzuki, 1999). In brief, AIC and BIC are both MDL with the difference in the complexity (penalty) term, while K2 is BDM (Bayesian Dirichlet Metric) due to the dirichlet distributional assumption, which makes it potentially not as robust as assumption-free MDL. These are realized in BNOmics as C++/ctypes modules. A novel entropy-based scoring function, similar to MDL, but more flexible with respect to the model complexity penalty, is also available, but it has not been extensively tested yet and will be described in detail elsewhere.

3.4. Output format and visualization

BNOmics outputs reconstructed BNs in DOT graph description markup language format. Subsequently, DOT files can be edited manually and visualized using Graphviz (open-source graph visualization software, http://www.graphviz.org). Graphviz can apply numerous network layouts to generate publication-quality illustrations. It is also possible to visualize smaller fragments of a BN (e.g., an immediate Markov neighborhood of a certain radius of a node of interest, such as phenotype), which is essential when visualizing BNs reconstructed from large datasets. In parallel, this allows Markov set-based classification (serving as both a variable selection routine and a classifier similar to Naïve Bayes).

4. RESULTS AND DISCUSSION

4.1. Example applications

Three applications below illustrate various aspects of BN analyses using BNOmics (namely exhaustive BN structure search, scalability, and visualization, and combining heterogeneous omics data and data types). The 2nd and 3rd applications relate to the Atherosclerosis Risk in Communities (ARIC), a comprehensive epidemiologic study of coronary heart disease (CHD), and its risk factors (ARIC investigators, 1989) and are detailed strictly from the analysis methodology prospective, with the biological results to be discussed elsewhere.

4.1.1. Variation in apolipoprotein E gene and plasma lipid and apolipoprotein E levels. This application illustrates exhaustive search in smaller BNs. Two datasets are included with the software distribution (available directly from the authors), both in self-explanatory .csv comma-delimited format and with SNP values recoded in 0,1,2,…format. These datasets were generated in a genetic epidemiology study of variation in the *apolipoprotein E (APOE)* gene and plasma lipid and APOE levels and were described in detail in Nickerson et al. (2000) and Stengård et al. (2002). Briefly, 20 SNPs in the *APOE* gene were genotyped in 702 African Americans from Jackson, Mississippi, and 854 non-Hispanic whites from Rochester, Minnesota. The datasets also contain plasma lipid and lipoprotein measurements and basic epidemiological variables. Three SNPs (designated #3937, #4075, and #4036 in the datasets) located in the coding region of the gene are associated with various phenotypes of interest. Importantly, SNPs, #3937 and #4075, code for the E2, E3, and E4 APOE isoforms. Plasma APOE level was the primary phenotype of interest in the original study. Therefore, the immediate Markov neighborhood of the BN node associated with the APOE plasma level variable received most scrutiny after the network reconstruction. The reconstructed BNs are shown in Figure 2 (Fig. 2a, African Americans; Fig. 2b, non-Hispanic whites). We refer the reader to Rodin et al. (2005) and Rodin et al. (2012) for the biological interpretation of the networks and will remark instead on the technical aspects of the network reconstruction and visualization.

The BNs in Figure 2 were reconstructed using MDL scoring metric. The continuous variables were discretized in three bins using entropy-based discretization with MDL stopping criterion. MDL scoring metric imposes a relatively high penalty on the model complexity, resulting in the reconstructed BNs being comparatively sparse (i.e., slightly underfitting). By varying the coefficients in the MDL/AIC metrics, degree of overfitting can be adjusted. The search continued until full exhaustion (bound by the machine epsilon, $\sim 2^{-53}$). The absence of any edges (hanging nodes at the top of Fig. 2a, b) indicates that the corresponding nodes are independent of other variables. A number next to the edge is proportional to the ratio of the model score of the BN with the edge to that of the BN without and quantifies the edge strength (also reflected in the edge color or thickness, which are some of the output options).

In both networks, the Markov neighborhoods of the node, APO_E, are consistent with what is known about the biology of ApoE and genetic epidemiology of *APOE* variation. In fact, the primary reason behind using these two datasets as a benchmark example application throughout the BNOmics development is that the APOE system can serve as a well-established true positive control with effects of known strength. As we change different algorithmic parameters and software settings, it is possible to study the robustness, sensitivity, and specificity of BN reconstruction by comparing the resulting APOE networks. For example, while similar BNs were obtained by us previously using different scoring metric and search engines (Rodin et al., 2005; Rodin et al., 2012), current BNs (Fig. 2) show improvement in robustness due to the exhaustive model selection process.

On a related note, we also applied BNOmics to the commonly used benchmark datasets, Alarm (Beinlich et al., 1989), Asia (Lauritzen and Spiegelhalter, 1988), and other well-known small datasets (see http://www.bnlearn.com/for repository), resulting in networks similar or identical to the actual BNs and BNs reconstructed by comparable software packages (results not shown).

4.1.2. GWAS analysis in ARIC study. This application demonstrates BNOmics' scalability in both BN reconstruction and visualization. The ARIC study is a population-based prospective cohort study of CHD and its risk factors (ARIC Investigators, 1989). ARIC recruited 15,792 non-Hispanic white and African American individuals aged 45–64 years at baseline (1987–89), chosen by probability sampling. The phenotypes of interest to the ARIC study include (among many others) CHD endpoint events (CHD

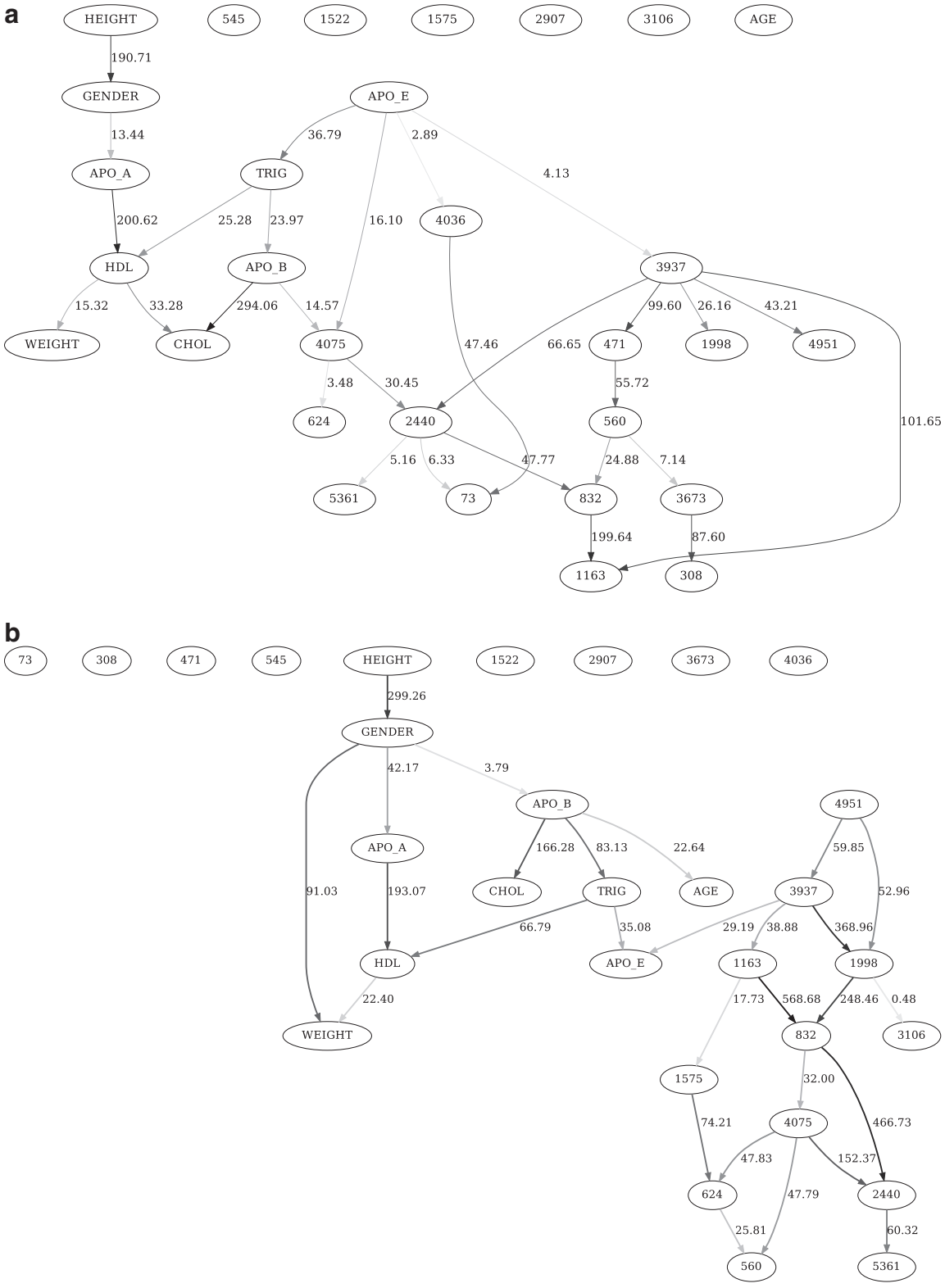

FIG. 2. BNs reconstructed from the APOE datasets. **(a)** African Americans from Jackson, Mississippi, **(b)** non-Hispanic whites from Rochester, Minnesota. Numbers next to BN edges indicate edge strengths. See text for interpretation of edge strength and disconnected notes. APO_E, APO_A, APO_B, TRIG, CHOL, and HDL stand for levels of apolipoproteins E, AI, and B, triglycerides, cholesterol, and high-density lipoprotein cholesterol, respectively. Number nodes indicate corresponding *APOE* SNPs. APOE, apolipoprotein E; SNP, single-nucleotide polymorphism.

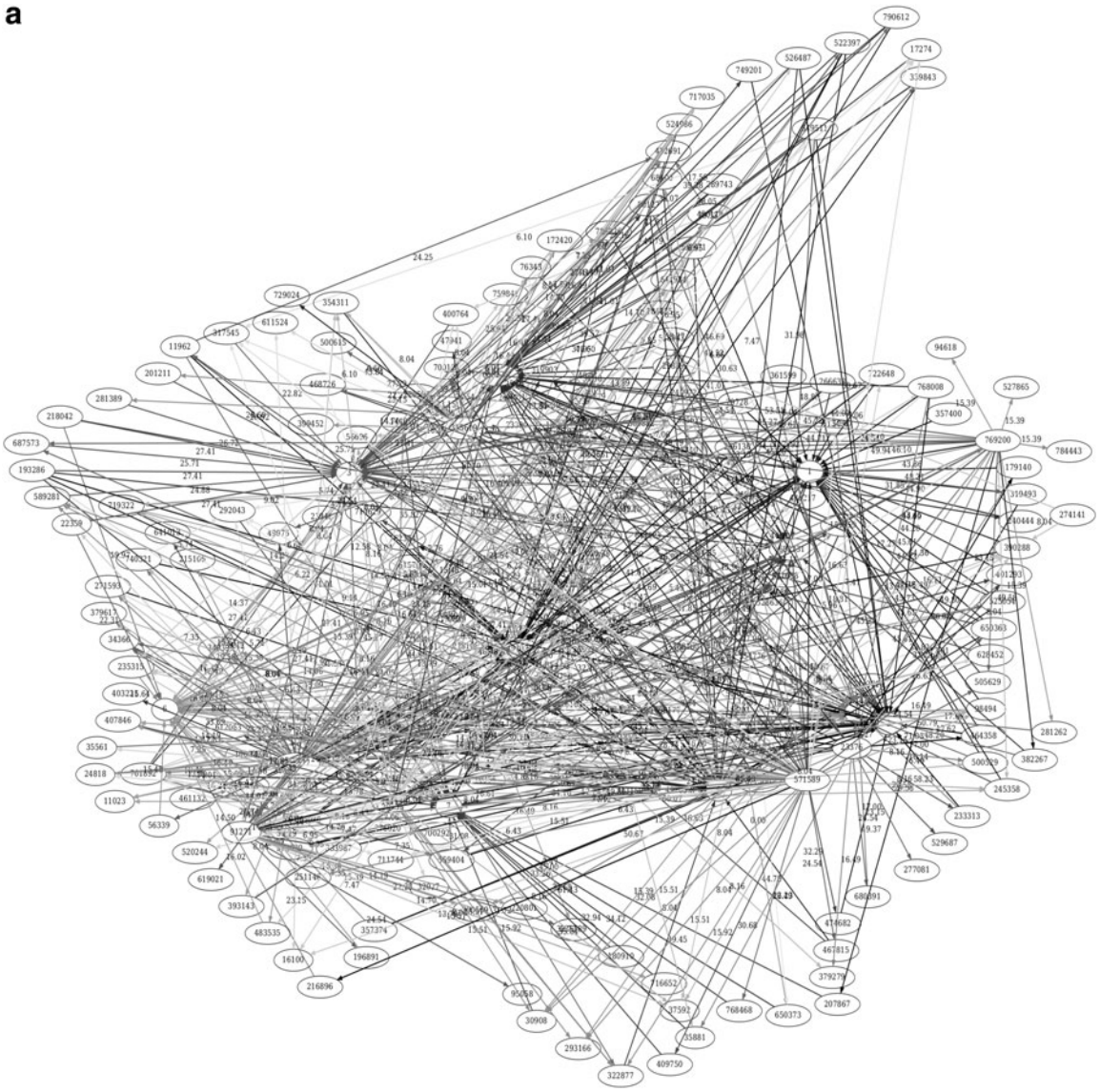

FIG. 3. **(a–c)** Visualization of the subnetworks of a BN reconstructed from the ARIC GWAS dataset. **(a)** Third-order (radius) Markov neighborhoods of blood lipid and epidemiological variables (nodes 1–8). Other number nodes correspond to the working SNP designations. Such fine scale does not permit for sensible visualization and is for methodology illustration purposes only. **(b)** Second-order (radius) Markov neighborhoods of blood lipid and epidemiological variables. **(c)** First-order (radius) Markov neighborhoods of blood lipid and epidemiological variables. Numbers next to BN edges indicate edge strengths. Sex, v1age01, hdl01, totchol, ldl02, trigs, bmi01, and glucos01 stand for gender, age, high-density lipoprotein cholesterol, total cholesterol, low-density lipoprotein cholesterol, triglycerides, BMI, and plasma glucose, respectively. Number nodes indicate corresponding SNPs. **(d)** BN reconstructed from eight non-SNP variables only, for comparison purposes. GWAS, genome-wide association study.

deaths, myocardial infarction, and hospitalized congestive heart failure), stroke events, metabolic syndrome, diabetes, blood pressure, and blood lipids. DNA samples ($\sim$900,000 SNPs genotyped using the Affymetrix 6.0 chip) have been collected on all members of the ARIC cohort.

When reconstructing BNs from the ARIC GWAS data, we were particularly interested in blood lipid phenotypes, thus concentrating on visualization of the subnetworks containing third-, second-, and first-order (radius) Markov neighborhoods of blood lipid variables (shown in Fig. 3a–c). It should be emphasized that the single reconstructed BN was built from all $\sim$900,000 SNP variables available—it is not shown for obvious reasons as third-order Markov neighborhood subnetwork is already almost impossible to visualize. Figure 3d shows a subnetwork containing non-SNP variables only.

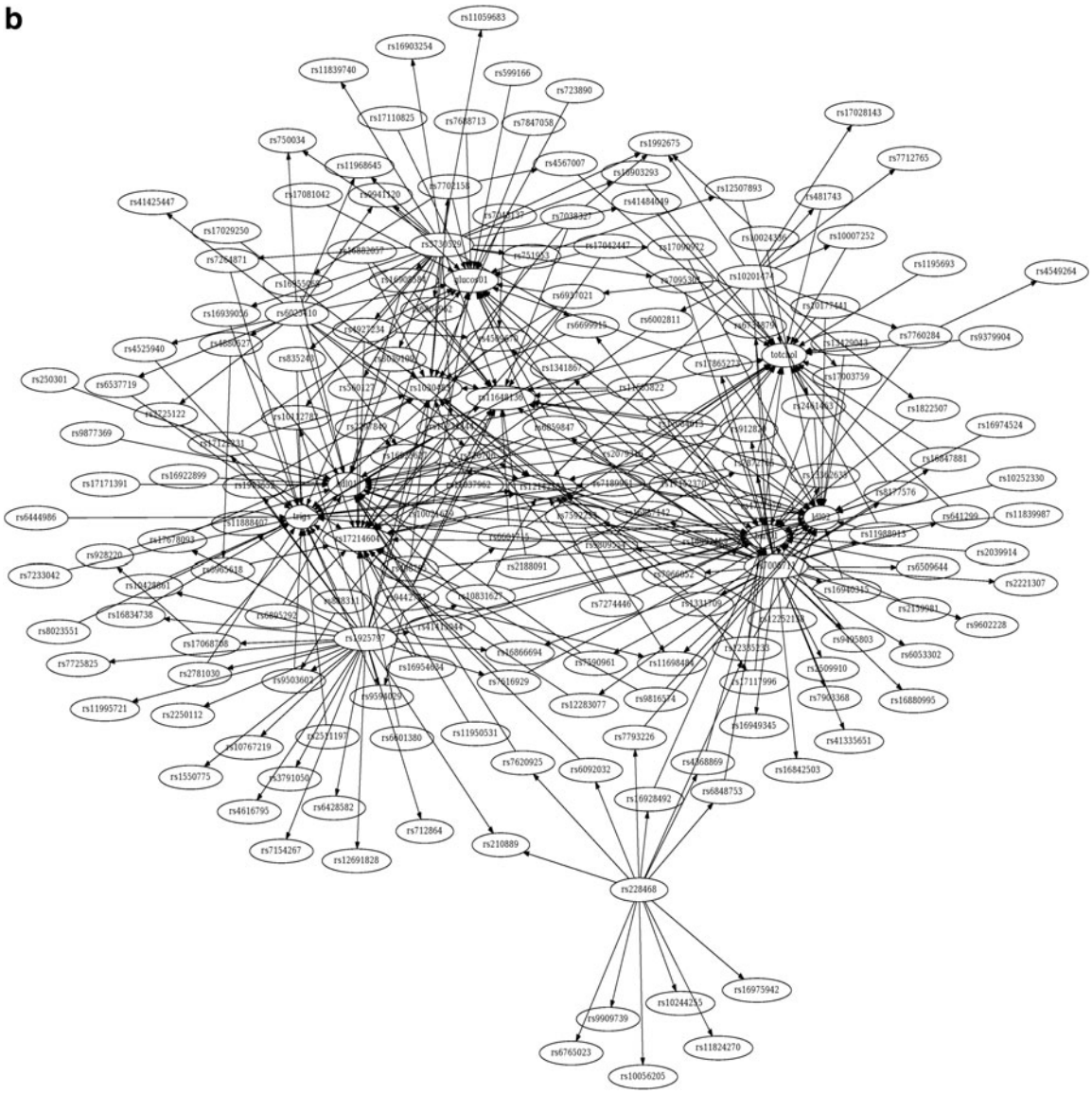

FIG. 3. (Continued).

4.1.3. Metabolomic analysis in ARIC study. This example illustrates BNOmics' ability to deal with heterogeneous (mixed) and unknown data types. One of the major challenges with data of this nature is that most variables are continuous with largely unknown (and different) distributions. Detailed description of metabolomic profile capture in a subset of ARIC study participants can be found in Zheng et al. (2014). Briefly, biochemical profiles of human sera ($\sim$600 metabolites) from $\sim$2000 ARIC African American individuals were obtained using current Metabolon platform (untargeted mass spectrometry). Approximately 350 known and $\sim$250 unknown metabolites were measured.

For the purposes of this application, primary phenotypes of interest were hypertension and heart failure (common in this subcohort). Analysis goals were to reconstruct the complete BN, to investigate the immediate network neighborhoods of hypertension and heart failure nodes, and to possibly assist in identifying the unknown metabolites; 208 reliably reproducible metabolites were included in the final BN analysis, together with relevant epidemiological variables. The full BN is shown in Figure 4a; Figure 4b depicts immediate (first-order) Markov neighborhood of hypertension phenotype variable. One unknown metabolite (X–11372) was indicated and is currently being followed up.

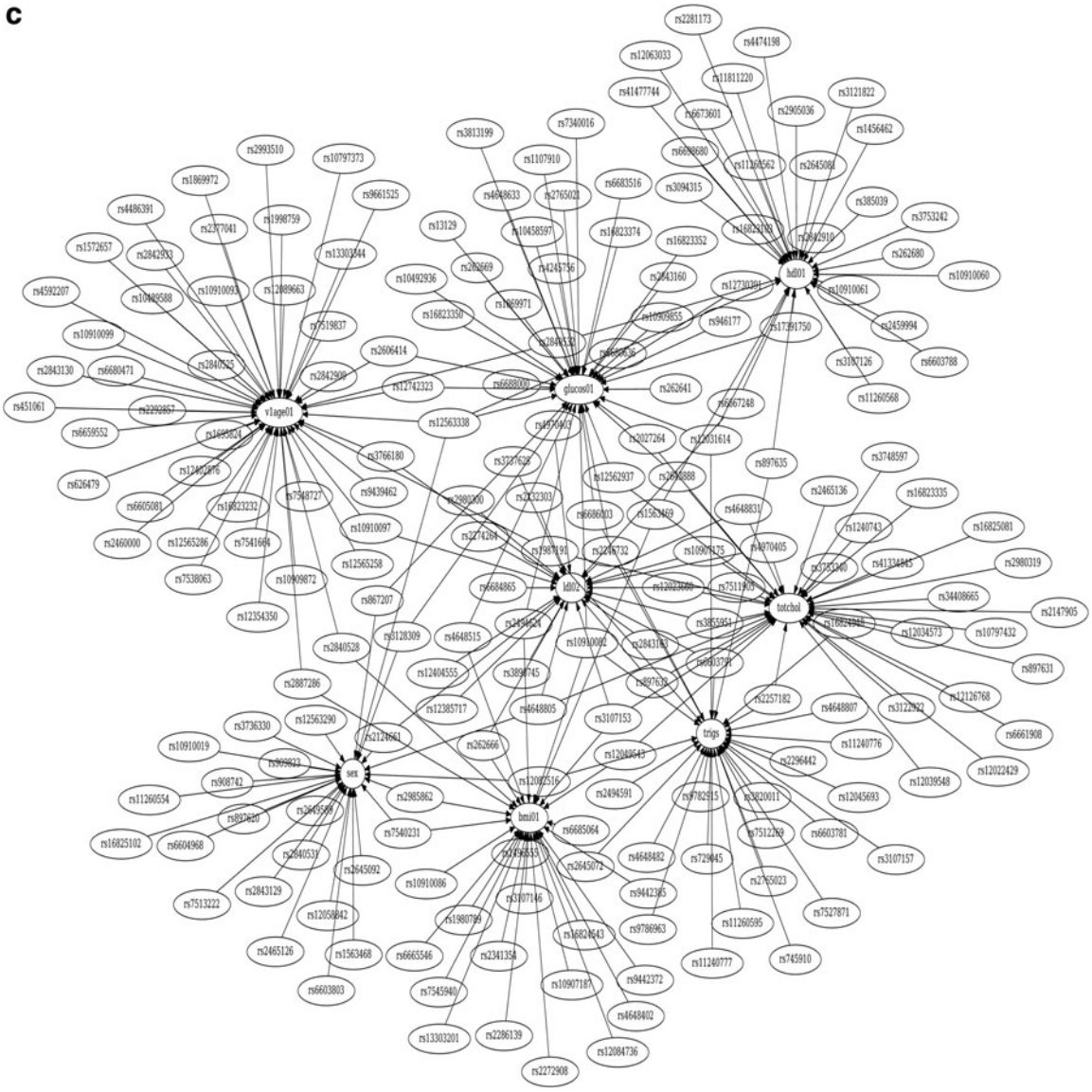

FIG. 3. (Continued).

4.2. *Scalability*

We have successfully applied BNOmics to various large-scale GWAS and metabolomic datasets, scalability being limited only by the hardware (memory) and preset computational time limits. Specifically, constructing a robust (converged) BN from a 900,000 SNP GWAS dataset took about 7 days on a regular 16GB 8-core workstation; a dataset with $\sim$100,000 variables required $\sim$27 hours, $\sim$10,000 variables required $\sim$10 hours (note that all of the efficiency improvements outlined above in the Data Storage and Input Format section were implemented in a manner specific to each dataset). Therefore, we suggest that potential users experiment with maxing out BNOmics on their respective hardware platforms. We also found scipy weave package (that allows inclusion of C/C++ into Python code) to be potentially useful when dealing with extremely large datasets.

At this time, there is only limited provision for parallelization, but an improved version of the algorithm that can take advantage of parallel computing is currently in the works. This becomes especially relevant with the number of samples $>10^6$. Currently, coarse multithreading/parallelization at the ancestor search level does not seem to provide significant benefit due to the fact that data passing to threads dominate the

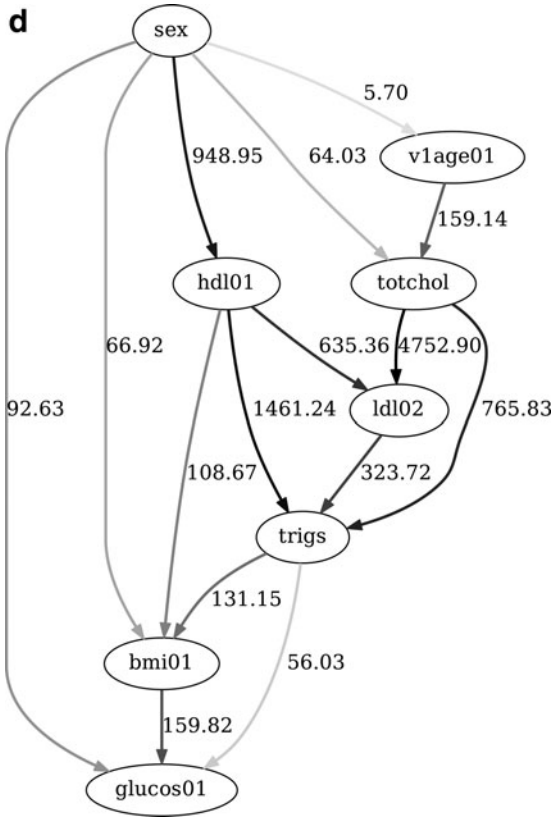

FIG. 3. (Continued).

process for large sample sizes, and the resulting process tends to be even less efficient than the serial version of the algorithm. A finer granularity level for multithreading/parallelization is necessary. Concurrently, in future, we plan to switch to the contiguous C arrays for improved data storage and retrieval. In general, we intend to recode the BNOmics prototype completely in C and maintain two versions—Python version for BN modeling research and methods testing and C version for the actual data analyses. This said, BNOmics is already more efficient than commonly used general-purpose BN reconstruction packages (e.g., bnlearn, http://www.bnlearn.com/and pebl, http://code.google.com/p/pebl-project/).

It should be reemphasized that while BNOmics outputs (in DOT language) a complete BN, for sufficiently large networks, it is impractical to deal with the output in its entirety (indeed, it makes no sense to generate a .pdf file visualizing a network with more than 100–200 nodes). Therefore, users are advised to concentrate on the smaller subnetworks by generating and visualizing lists of parents and children (i.e., immediate ancestors and descendants, respectively) of certain nodes of interest, such as phenotypes (see typical examples in preceding section), or subgraphs defined by the order/radius of relationship (a certain number of generations up and down family tree). In future, we plan to extend this feature by adding the adjacency matrix representation of the ancestor–descendant relationships.

4.3. Future directions

We are committed to the continuing open-source flexible code development of BNOmics. While the basic modular organization of the package will remain intact, we plan to introduce major improvements in usability and efficiency in response to the increasing availability of very large-scale datasets (that are presently being generated as part of our ongoing research projects). We are also soliciting ideas and suggestions for improvements from the greater community of potential users, with a special emphasis on (but certainly not limited to) automated format conversion for diverse datasets. We are currently carrying out the analyses of epigenetic (methylome) and immune system datasets, to name just a couple. Another area of future concentration is using BNOmics to vary and compare, in the context of comprehensive

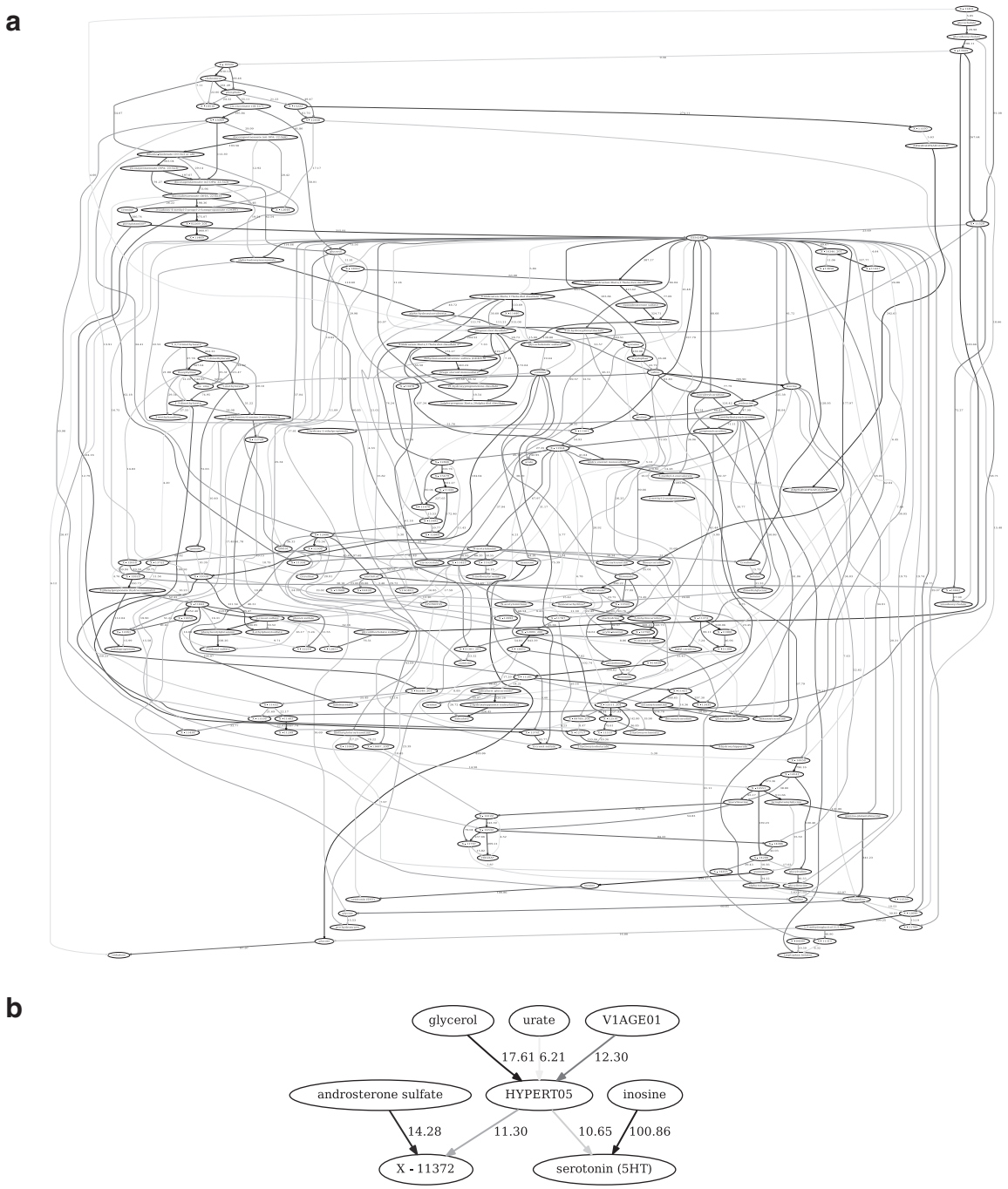

FIG. 4. **(a)** BN reconstructed from the ARIC metabolomic profile dataset. **(b)** Visualization of a first-order (radius) Markov neighborhood subnetwork of hypertension phenotype node (HYPERT05). Numbers next to BN edges indicate edge strengths. Epidemiological and known metabolite node designations are largely self-explanatory (e.g., V1AGE01, glycerol). X—<…> nodes indicate unknown metabolites. See Zheng et al., 2014, for more detail.

simulation studies, different BN reconstruction alternatives with respect to scoring functions/metrics, local search/optimization algorithms, and discretization procedures. The Python code was developed from the very beginning with such simulation studies in mind as various changes in the reconstruction scheme can be incorporated on the fly. In this sense, BNOmics can be used not only as a data analysis tool but also as a platform for the investigation, and improvement, of different aspects of the BN modeling process. Work currently in progress is itemized in Table 1.

Table 1. Work on the Algorithm/Software Improvements Currently in Progress

Developing minimal API, GUI, web server, and context help
Automated dataset conversion for common and emerging data formats
Fine parallelization of core algorithm
Implementation and investigation of higher-order search algorithms
Switching to C arrays for optimized data storage and retrieval
Switching to adjacency matrix representation of ancestor–descendant relationships
Incorporation of predefined BN structure priors (available now, but only manually)
Testing of a novel entropy-based scoring function/metric (cumulative entropy) and improving treatment of continuous variables (which right now is less efficient than that of discrete variables)
Testing a network neighborhood-based imputation algorithm
Implementation of fast Markov set-based classifier

5. CONCLUSIONS

Three features of BNOmics set it apart from comparable alternatives—first, its high computational efficiency and scalability; second, flexibility and open nature of the source code; and third, its immediate applicability to the large-scale datasets generated by the omics studies. BNOmics has been very useful in our own research projects and collaborations. Currently, there is a substantial interest in applying systems biology thinking and analysis methods to the large-scale omics data (Qi et al., 2014; Agostinho et al., 2015; Sherif et al., 2015; Marini et al., 2015; Yin et al., 2015b; Kaiser et al., 2016). However, the assortment of workable systems biology data analysis tools is very limited especially if the ultimate goal is reverse engineering of biological networks from the massive flat datasets. BN is a useful paradigm for biological network reconstruction and modeling, and BNOmics is a powerful implementation thereof.

6. AVAILABILITY AND REQUIREMENTS

Project name: BNOmics

Project home page: TBA (City of Hope), available directly from the authors or at Bitbucket (baseline implementation)

Operating system(s): Source code is available for any standard Python implementation

Programming language: Python, C/C++

Other requirements: Python interpreter, linear algebra libraries (numpy), and optional C/C++ code inclusion libraries (scipy)

License: Worldwide nonexclusive, standard open-source

ACKNOWLEDGMENTS AND FUNDING

The authors are grateful to Anatolii Litvinenko for his earlier work on BN reconstruction. The authors would also like to thank D.C. Rao and Sergio Branciamore for insightful suggestions regarding BN reconstruction and Arthur D. Riggs and Peter P. Lee for many useful discussions on BN modeling in the context of newly emerging data. A.S.R. holds the Susumu Ohno Chair in Theoretical Biology at Beckman Research Institute of the City of Hope. E.B. is supported by NIH grants, 5U01AG049506 and 5R01NS087541. This study was also partially supported by grants from PhRMA foundation and NIH to A.S.R. and E.B.

AUTHORS' CONTRIBUTIONS

A.S.R. drafted the manuscript, conceived the study, and contributed to implementing the algorithm and analyzing the data. G.G. implemented the algorithm, analyzed the data, and contributed to drafting the manuscript and conceiving the study. E.B. collected and interpreted the data and contributed to drafting the manuscript, conceiving the study, and analyzing the data.

AUTHOR DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

Agostinho, N.B., Machado, K.S., and Werhli, A.V. 2015. Inference of regulatory networks with a convergence improved MCMC sampler. *BMC Bioinformatics* 16, 306.

Akaike, H. 1974. A new look at the statistical identification problem. *IEEE Trans. Auto. Control* 19, 716–723.

ARIC Investigators. 1989. The Atherosclerosis Risk in Communities (ARIC) study: Design and objectives. *Am. J. Epidemiol.* 129, 687–702.

Beinlich, I.A., Suermondt, H.J., Chavez, R.M., et al. 1989. The ALARM monitoring system: A case study with two probablistic inference techniques for belief networks. Second European Conference on Artificial Intelligence in Medicine, London, 38, 247–256.

Chickering, D.M., Heckerman, D., and Meek, C. 2004. Large-sample learning of Bayesian networks is NP-hard. *J. Mach. Learn. Res*. 5, 1278–1330.

Cho, H., Berger, B., and Peng, J. 2016. Reconstructing causal biological networks through active learning. *PLoS One* 11, e0150611.

Clarke, E.J., and Barton, B.A. 2000. Entropy and MDL discretization of continuous variables for bayesian belief networks. *Int. J. Intell. Syst*. 15, 61–92.

Cooper, G.F., and Herskovits, E. 1992. A Bayesian method for the induction of probabilistic networks from data. *Mach. Learn*. 9, 309–347.

Djebbari, A., and Quackenbush, J. 2008. Seeded Bayesian networks: Constructing genetic networks from microarray data. *BMC Syst. Biol*. 2, 57.

Fayyad, U., and Irani, K. 1993. Multi-interval discretization of continuous-valued attributes for classification learning. Proceedings of the 13th International Joint Conference on Artificial Intelligence. San Mateo, CA: Morgan Kaufmann; Vol. 2, 1022–1027.

Friedman, N., Nachman, I., and Pe'er, D. 1999. Learning Bayesian network structure from massive datasets: The ''sparse candidate'' algorithm. Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI'99). 196–205. Morgan Kaufman, Burlington, MA.

Friedman, N., Linial, M., Nachman, I., et al. 2000. Using Bayesian networks to analyze expression data. *J. Comput. Biol*. 7, 601–620.

Gao, T., and Ji, Q. 2016. Efficient Markov blanket discovery and its application. *IEEE Trans. Cybern.* [Epub ahead of print]. PMID: 27046886.

Guan, D., Shao, J., Deng, Y., et al. 2014. CMGRN: A web server for constructing multilevel gene regulatory networks using ChIP-seq and gene expression data. *Bioinformatics.* PMID: 24389658.

Han, B., Chen, X.W., Talebizadeh, Z., et al. 2012. Genetic studies of complex human diseases: Characterizing SNP-disease associations using Bayesian networks. *BMC Syst. Biol.* 6 Suppl 3, S14.

Heckerman, D.A. 1995. Tutorial on Learning with Bayesian Networks. Technical Report MSR-TR-95–06, Microsoft Research.

Jiang, X., Barmada, M.M., and Visweswaran, S. 2010a. Identifying genetic interactions in genome-wide data using Bayesian networks. *Genet. Epidemiol*. 34, 575–581.

Jiang, X., Jao, J., and Neapolitan, R. 2015. Learning predictive interactions using information gain and Bayesian network scoring. *PLoS One* 10, e0143247.

Jiang, X., and Neapolitan, R.E. 2012. Mining pure, strict epistatic interactions from high-dimensional datasets: Ameliorating the curse of dimensionality. *PLoS One* 7, e46771.

Jiang, X., and Neapolitan, R.E. 2015. Evaluation of a two-stage framework for prediction using big genomic data. *Brief Bioinform.* 16, 912–921.

Jiang, X., Neapolitan, R.E., Barmada, M.M., et al. 2010b. A fast algorithm for learning epistatic genomic relationships. *AMIA Annu. Symp. Proc.* 2010, 341–345.

Kaiser, J.L., Bland, C.L., and Klinke, D.J., 2nd. 2016. Identifying causal networks linking cancer processes and anti-tumor immunity using Bayesian network inference and metagene constructs. *Biotechnol. Prog.* 32, 470–479.

Keilwagen, J., Grau, J., Posch, S., et al. 2010. Apples and oranges: Avoiding different priors in Bayesian DNA sequence analysis. *BMC Bioinformatics* 11, 149.

Lauritzen, S.L., and Spiegelhalter, D.J. 1988. Local computations with probabilities on graphical structures and their application to expert systems. *J. R. Stat. Soc. Series B* 50, 157–224.

Li, R., Dudek, S.M., Kim, D., et al. 2016. Identification of genetic interaction networks via an evolutionary algorithm evolved Bayesian network. *BioData Min.* 9, 18.

Liu, C.C., Tseng, Y.T., Li, W., et al. 2014. DiseaseConnect: A comprehensive web server for mechanism-based disease-disease connections. *Nucleic Acids Res*. 42, W137–W146.

Lo, L.Y., Wong, M.L., Lee, K.H., et al. 2015. High-order dynamic Bayesian Network learning with hidden common causes for causal gene regulatory network. *BMC Bioinformatics* 16, 395.

Marini, S., Trifoglio, E., Barbarini, N., et al. 2015. A Dynamic Bayesian Network model for long-term simulation of clinical complications in type 1 diabetes. *J. Biomed. Inform.* 57, 369–376.

Neapolitan, R., Xue, D., and Jiang, X. 2014. Modeling the altered expression levels of genes on signaling pathways in tumors as causal bayesian networks. *Cancer Inform.* 13, 77–84.

Neapolitan, R.E. 2004. *Learning Bayesian Networks*. Prentice Hall, Upper Saddle River, NJ.

Nickerson, D.A., Taylor, S.L., Fullerton, S.M., et al. 2000. Sequence diversity and large-scale typing of SNPs in the human apolipoprotein E gene. *Genome Res.* 10, 1532–1545.

Nielsen, J., and Mailund, T. 2008. SNPFile—A software library and file format for large scale association mapping and population genetics studies. *BMC Bioinformatics* 9, 526.

Paluszewski, M., and Hamelryck, T. 2010. Mocapy++—A toolkit for inference and learning in dynamic Bayesian networks. *BMC Bioinformatics* 11, 126.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems*. San Mateo, CA, Morgan Kaufmann.

Pearl, J. 2000. *Causality: Models, Reasoning, and Inference*, Cambridge University Press, Cambridge, UK.

Pe'er, D. 2005. Bayesian network analysis of signaling networks: A primer. *Sci. STKE*. 2005, l4.

Qi, Q., Li, J., and Cheng, J. 2014. Reconstruction of metabolic pathways by combining probabilistic graphical model-based and knowledge-based methods. *BMC Proc.* 8 (Suppl 6 Proceedings of the Great Lakes Bioinformatics Confer):S5.

Rissanen, J. 1978. Modeling by shortest data description. *Automatica*. 14, 465–471.

Rodin, A., Mosley, T.H. Jr., Clark, A.G., et al. 2005. Mining genetic epidemiology data with Bayesian networks application to APOE gene variation and plasma lipid levels. *J. Comput. Biol*. 12, 1–11.

Rodin, A.S., Gogoshin, G., Litvinenko, A., et al. 2012. Exploring genetic epidemiology data with bayesian networks, 479–510. *In Handbook of Statistics*, Vol. 28. Eds: R. Chakraborty, C.R. Rao, and P.K. Sen. Elsevier B.V.

Schwarz, G. 1978. Estimating the dimension of a model. *Ann. Stat*. 6, 461–464.

Sherif, F.F., Zayed, N., and Fakhr, M. 2015. Discovering Alzheimer genetic biomarkers using Bayesian networks. *Adv. Bioinformatics* 2015, 639367.

Steele, E., Tucker, A., 't Hoen, P.A., et al. 2009. Literature-based priors for gene regulatory networks. *Bioinformatics* 25, 1768–1774.

Stengård, J.H., Clark, A.G., Weiss, K.M., et al. 2002. Contributions of 18 additional DNA sequence variations in the gene encoding apolipoprotein E to explaining variation in quantitative measures of lipid metabolism. *Am. J. Hum. Genet*. 71, 501–517.

Suzuki, J. 1999. Learning Bayesian belief networks based on the minimum description length principle: Basic properties. *IEICE Trans. Fundam.* E82–A, 2237–2245.

Tasaki, S., Sauerwine, B., Hoff, B., et al. 2015. Bayesian network reconstruction using systems genetics data: Comparison of MCMC methods. *Genetics* 99, 973–989.

Vass, J.K., Higham, D.J., Mudaliar, M.A., et al. 2011. Discretization provides a conceptually simple tool to build expression networks. *PLoS One* 6, e18634.

Watcher, A., and Beisbarth, T. 2015. pwOmics: An R package for pathway-based integration of time-series omics data using public database knowledge. *Bioinformatics* 31, 3072–3074.

Yin, W., Garimalla, S., Moreno, A., et al. 2015a. A tree-like Bayesian structure learning algorithm for small-sample datasets from complex biological model systems. *BMC Syst. Biol.* 9, 49.

Yin, W., Kissinger, J.C., Moreno, A., et al. 2015b. From genome-scale data to models of infectious disease: A Bayesian network-based strategy to drive model development. *Math. Biosci*. 270(Pt B), 156–168.

Zhang, X., Xue, F., Liu, H., et al. 2014. Integrative Bayesian variable selection with gene-based informative priors for genome-wide association studies. *BMC Genet.* 15, 130.

Zheng, Y., Yu, B., Alexander, D., et al. 2014. Metabolomic patterns and alcohol consumption in African Americans in the Atherosclerosis Risk in Communities Study. *Am. J. Clin. Nutr.* 99, 1470–1478.

Ziebarth, J.D., Bhattacharya, A., and Cui, Y. 2013. Bayesian Network Webserver: A comprehensive tool for biological network modeling. *Bioinformatics*. 29, 2801–2803.

Address correspondence to:
Dr. Andrei S. Rodin
Diabetes and Metabolism Research Institute
City of Hope
1500 East Duarte Road
Duarte, CA 91010

E-mail: arodin@coh.org